



Gesellschaft für Operations Research e.V. (GOR)

D. Ahr · R. Fahrion
M. Oswald · G. Reinelt
Editors

Operations Research Proceedings 2003



Springer

Operations Research Proceedings 2003

Selected Papers
of the International Conference
on Operations Research (OR 2003)

Heidelberg, September 3-5, 2003

Springer-Verlag Berlin Heidelberg GmbH

D. Ahr · R. Fahrion
M. Oswald · G. Reinelt
Editors

Operations Research Proceedings 2003

Selected Papers
of the International Conference
on Operations Research (OR 2003)

Heidelberg, September 3-5, 2003

With 137 Figures
and 51 Tables



Springer

Dipl.-Inf. Dino Ahr
Universität Heidelberg
Institut für Informatik
Im Neuenheimer Feld 368
69120 Heidelberg, Germany
Dino.Ahr@informatik.uni-heidelberg.de

Dr. Marcus Oswald
Universität Heidelberg
Institut für Informatik
Im Neuenheimer Feld 368
69120 Heidelberg, Germany
Marcus.Oswald@informatik.uni-heidelberg.de

Professor Dr. Roland Fahrion
Universität Heidelberg
Alfred-Weber-Institut
Grabengasse 14
69117 Heidelberg, Germany
Roland.Fahrion@awi.uni-heidelberg.de

Professor Dr. Gerhard Reinelt
Universität Heidelberg
Institut für Informatik
Im Neuenheimer Feld 368
69120 Heidelberg, Germany
Gerhard.Reinelt@informatik.uni-heidelberg.de

ISBN 978-3-540-21445-8 ISBN 978-3-642-17022-5 (eBook)
DOI 10.1007/978-3-642-17022-5

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek

Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data available in the internet at <http://dnb.ddb.de>

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Originally published by Springer-Verlag Berlin Heidelberg New York in 2004

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: Erich Kirchner, Heidelberg

SPIN 10998311 42/3130 – 5 4 3 2 1 0 – Printed on acid-free paper

Preface

This volume contains a selection of papers referring to lectures presented at the symposium “Operations Research 2003” (OR03) held at the Ruprecht-Karls-Universität Heidelberg, September 3 – 5, 2003. This international conference took place under the auspices of the German Operations Research Society (GOR) and of Dr. Erwin Teufel, prime minister of Baden-Württemberg.

The symposium had about 500 participants from countries all over the world. It attracted academicians and practitioners working in various field of Operations Research and provided them with the most recent advances in Operations Research and related areas in Economics, Mathematics, and Computer Science.

The program consisted of 4 plenary and 13 semi-plenary talks and more than 300 contributed papers selected by the program committee to be presented in 17 sections.

Due to a limited number of pages available for the proceedings volume, the length of each article as well as the total number of accepted contributions had to be restricted. Submitted manuscripts have therefore been reviewed and 62 of them have been selected for publication. This refereeing procedure has been strongly supported by the section chairmen and we would like to express our gratitude to them.

Finally, we also would like to thank Dr. Werner Müller from Springer-Verlag for his support in publishing this proceedings volume.

Heidelberg, February 2004

Dino Ahr
Roland Fahrion
Marcus Oswald
Gerhard Reinelt

Committees

Organizing Committee

Roland Fahrion (Heidelberg)

Gerhard Reinelt (Heidelberg)

Program Committee

Hans-Georg Bock (Heidelberg)

Wolfgang Domschke (Darmstadt)

Roland Fahrion (Heidelberg)

Michael Jünger (Köln)

Hartmut Kogelschatz (Heidelberg)

Gerhard Reinelt (Heidelberg)

Günter D. Liesegang (Heidelberg)

Franz Rendl (Klagenfurt)

Gerhard Wäscher (Magdeburg)

Sections and Chairpersons

Revenue Management

Robert Klein (Darmstadt), Gerhard Wäscher (Magdeburg)

Telecommunication and Information Technology

Alexander Martin (Darmstadt)

Production, Logistics and Supply Chain Management

Bernhard Fleischmann (Augsburg)

Services, Transportation and Traffic

Tore Grünert (Aachen)

Scheduling and Project Management

Andreas Drexl (Kiel)

Marketing and Data Analysis

Wolfgang Gaul (Karlsruhe)

Energy, Environment and Health

Steffen Fleßa (Heidelberg), Günter D. Liesegang (Heidelberg)

Finance, Banking and Insurances

Siegfried Trautmann (Mainz)

Simulation

Hans-Otto Günther (Berlin)

Continuous Optimization

Johannes Jahn (Erlangen)

Discrete and Combinatorial Optimization

Peter Gritzmann (München)

Applied Probability

Ulrich Rieder (Ulm)

Artificial Intelligence, Fuzzy Logic and Neural Networks

Thorsten Poddig (Bremen), Heinrich Rommelfanger (Frankfurt/M.)

Econometrics, Statistics, Mathematical Economics and Decision Theory

Stefan Huschens (Dresden)

Experimental Economics, Game Theory and Auctioning

Marco Lehmann-Waffenschmidt

Managerial Accounting

Christian Hofmann (Hannover), Carsten Homburg (Köln)

Web Technology, Knowledge Management and Decision Support Systems

Leena Suhl (Paderborn)

Contents

GOR Awards

- Assessing Capacity Improvements by Relaying in Cellular Networks** 1
H.-F. Geerdes
- Performance Analysis of *M*-designed Inbound Call Centers** 9
R. Stolletz

Revenue Management

- Revenue Management in Manufacturing** 17
F. Defregger, H. Kuhn
- Network Revenue Management: Some Issues on Upper and Lower Bounds** 23
M. Müller-Bungart

Telecommunication and Information Technology

- Optimisation Methods for UMTS Radio Network Planning** 31
A. Eisenblätter, A. Fügenschuh, H.-F. Geerdes, D. Junglas, T. Koch, A. Martin

Production, Logistics and Supply Chain Management

- Distribution Planning Problem: A Survey** 39
B. Bilgen, I. Ozkarahan
- Artikelanordnungsmuster bei Mann-zur-Ware-Kommissionierung** 47
K. Dörner, M. Reeh, C. Strauss, G. Wäscher
- The Impact of the Exchange of Market and Stock Information on the Bullwhip Effect in Supply Chains** 55
B. Faißt, D. Arnold, K. Furmans

Analyzing the Bullwhip Effect of Installation-Stock and Echelon-Stock Policies with Linear Control Theory	63
K. Hoberg, U.W. Thonemann, J.R. Bradley	
Policy Approximation for the Production Inventory Problem with Stochastic Demand, Stochastic Yield and Production Leadtime	71
K. Inderfurth, C. Gotzel	
A Dynamic Model for Choosing the Optimal Technology in the Context of Reverse Logistics	79
R. Kleber	
Deriving Inventory-Control Policies for Periodic Review with Genetic Programming	87
P. Kleinau, U.W. Thonemann	
Dynamic Multi-Commodity Facility Location: A Mathematical Modeling Framework for Strategic Supply Chain Planning	95
M.T. Melo, S. Nickel, F. Saldanha da Gama	
Leistungsabstimmung von Produktionslinien in der Elektronikmontage	103
M. Schleusener, H.-O. Günther	
A Priority-Rule Based Method for Batch Production Scheduling in the Process Industries	111
C. Schwindt, N. Trautmann	
Services, Transportation and Traffic	
A Metaheuristic Approach for Hazardous Materials Transportation	119
P. Carotenuto, G. Galiano, S. Giordani	
Personal- und Fahrzeugeinsatzplanung in der Müllentsorgung	127
J.R. Daduna	
Modelling of Complex Costs and Rules in a Crew Pairing Column Generator	133
R. Galia, C. Hjørring	

Convexification of the Traffic Equilibrium Problem with Social Marginal Cost Tolls	141
P. O. Lindberg, L. Engelson	

Vermittlung von Fahrgemeinschaften betrachtet als Vehicle Routing Problem	149
G. Reents	

Scheduling and Project Management

Single Machine Scheduling with Precedence Constraints and SLK Due Date Assignment	157
V. Gordon, J.-M. Proth, V. Strusevich	

A Parallel Approach to the Pricing Step in Crew Scheduling Problems	165
T.V. Hoai, G. Reinelt, H.G. Bock	

Scheduling Regular and Temporary Employees with Qualifications in a Casino	173
C. Stark, J. Zimmermann	

Marketing and Data Analysis

Web Robot Detection - the Influence of Robots on Web Mining	181
C. Bomhardt, W. Gaul	

Visualizing Recommender System Results via Multidimensional Scaling	189
W. Gaul, P. Thoma, L. Schmidt-Thieme, S. van den Bergh	

Die Modellierung von Präferenzveränderungen mittels Scanner Panel Daten	197
L. Hildebrandt, L. Michaelis	

Measurement of Online Visibility	205
N. Schmidt-Mänz, W. Gaul	

Determinants and Behavioral Consequences of Customer Loyalty and Dependence in Online Brokerage - Results from a Causal Analysis	213
Y. Staack	

Product Bundling as a Marketing Application	221
B. Stauß, W. Gaul	

Energy, Environment and Health

Entwicklung und Anwendung einer mehrstufigen Methodik zur Analyse betriebsübergreifender Energieversorgungskonzepte	229
W. Fichtner, O. Rentz	

A System Dynamics Model of the Epidemiological Transition	237
S. Fleßa	

Finance, Banking and Insurances

Implementing a Reference Portfolio Strategy in Bond Portfolio Management	245
U. Derigs, N.-H. Nickel	

Evolutionary Algorithms and the Cardinality Constrained Portfolio Optimization Problem	253
F. Streichert, H. Ulmer, A. Zell	

Calculating Concentration-Sensitive Capital Charges with Conditional Value-at-Risk	261
D. Tasche, U. Theiler	

Simulation

Integration der Simulation in die Programmplanung einer globalen Supply Chain	269
L. Dohse, T. Hanschke, I. Meents, H. Zisgen	

Objektorientierte Simulation von Anlagen der Elektronikmontage	276
M. Grunow, H.-O. Günther	

Continuous Optimization

A Mixed-Discrete Bilevel Programming Problem	284
S. Dempe, V. Kalashnikov	

Detecting Superfluous Constraints in Quadratic Programming by Varying the Optimal Point of the Unrestricted Problem	292
--	------------

P. Recht

Fast Optimal Control Algorithms with Application to Chemical Engineering	300
---	------------

A. Schäfer, U. Brandt-Pollmann, M. Diehl, H.G. Bock, J.P. Schlöder

Discrete and Combinatorial Optimization

Proofs of Unsatisfiability Via Semidefinite Programming	308
--	------------

M. Anjos

Algorithms with Performance Guarantees for a Metric Problem of Finding Two Edge-Disjoint Hamiltonian Circuits of Minimum Total Weight	316
--	------------

A.E. Baburin, E.K. Gimadi, N.M. Korkishko

A Quadratic Optimization Model for the Consolidation of Farmland by Means of Lend-Lease Agreements	324
---	------------

A. Brieden, P. Gritzmann

A Bipartite Graph Simplex Method	332
---	------------

R. Euler

Regions of Stability for Nonlinear Discrete Optimization Problems	340
--	------------

D. Fanghänel

Optimization Models for the Containership Stowage Problem	347
--	------------

P. Giemsch, A. Jellinghaus

Solving the Sequential Ordering Problem with Automatically Generated Lower Bounds	355
--	------------

I.T. Hernádvölgyi

Scheduling Jobs with a Stepwise Function of Change of Their Values	363
---	------------

A. Janiak, A. Kasperski, T. Krysiak

Small Instance Relaxations for the Traveling Salesman Problem	371
--	------------

G. Reinelt, K.M. Wenger

Applied Probability

A Remark on Multiobjective Stochastic Optimization Problems: Stability and Empirical Estimates	379
---	------------

V. Kaňková

Portfolio Optimization under Partial Information: Stochastic Volatility in a Hidden Markov Model	387
---	------------

J. Sass, U.G. Haussmann

On the Set of Optimal Policies in Variance Penalized Markov Decision Chains	395
--	------------

K. Sladký, M. Sitař

On Random Sums and Compound Process Models in Financial Mathematics	403
--	------------

P. Volf

Artificial Intelligence, Fuzzy Logic and Neural Networks

Structural Optimization in Aircraft Engineering using Support Vector Machines	411
--	------------

P. Kaletta, K. Wolf, A. Fischer

Zeitformen in einer probabilistischen Konditionallogik	419
---	------------

E. Reucher, W. Rödder

Dienstplanbewertung mit unscharfen Regeln	427
--	------------

A. Schroll, T. Spengler

Optimization by Gaussian Processes assisted Evolution Strategies	435
---	------------

H. Ulmer, F. Streichert, A. Zell

Econometrics, Statistics, Mathematical Economics and Decision Theory

Stochastic Programming and Statistical Estimates	443
---	------------

S. Vogel

Time Lags in Capital Accumulation	451
--	------------

R. Winkler, U. Brandt-Pollmann, U. Moslener, J.P. Schlöder

Experimental Economics, Game Theory and Auctioning

First-Price Bidding and Entry Behavior in a Sequential Procurement Auction Model 459

J. P. Reiß, J. R. Schöndube

Web Technology, Knowledge Management and Decision Support Systems

Financial Market Web Mining with the Software Agent PISA 467

P. Bartels, M.H. Breitner

Non-Linear Programming Solvers for Decision Analysis 475

X. Ding, M. Danielson, L. Ekenberg

knowCube - A Spreadsheet Method for Interactive Multicriteria Decision Making 483

H.L. Trinkaus

Assessing Capacity Improvements by Relaying in Cellular Networks*

Hans-Florian Geerdes

Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany

Abstract. Relaying – allowing multiple wireless hops – is a protocol extension for cellular networks conceived to improve data throughput. Its benefits have only been quantified for small example networks. For assessing its general potential, we define a complex resource allocation/scheduling problem. Several mathematical models are presented for this problem; while a time-expanded MIP approach turns out intractable, a sophisticated column generation scheme leads to good computational results. We thereby show that for selected cases relaying can increase data throughput by 30% on the average.

1 Introduction

The amount of data that can be sent per time in wireless computer networks (wireless LANs, WLANs) is limited by bandwidth restrictions and by interference. Many researchers strive to use the available resources more efficiently in order to increase data throughput. One technique conceived to do so is *relaying*.

1.1 Relaying in Cellular Networks

Wireless telecommunication networks fall into two categories: *ad hoc* networks (Fig. 1(a)) and *cellular* networks (Fig. 1(b)). In *ad hoc* networks, the mobile nodes organize their usage of the radio network themselves, whereas in cellular networks this task is carried out by an infrastructure of fixed *base stations*. In contrast to *ad hoc* networks, this implies that mobile nodes may never exchange data directly, they are only allowed to transmit data to and receive data from their respective base station.

Relaying means to allow multiple wireless hops for data to reach its destination. While this is quite common in most *ad hoc* networks, it is normally not used in cellular ones. To use relaying in cellular computer networks can be beneficial because transmission speed can be adapted – several fast hops might be better on the overall than one slow one. Within this work, however, we still rely on the the main feature of cellular network organization, namely organization in space and in time by the base station infrastructure.

* Supported by the DFG research center “Mathematics for key technologies” (FZT 86) in Berlin.

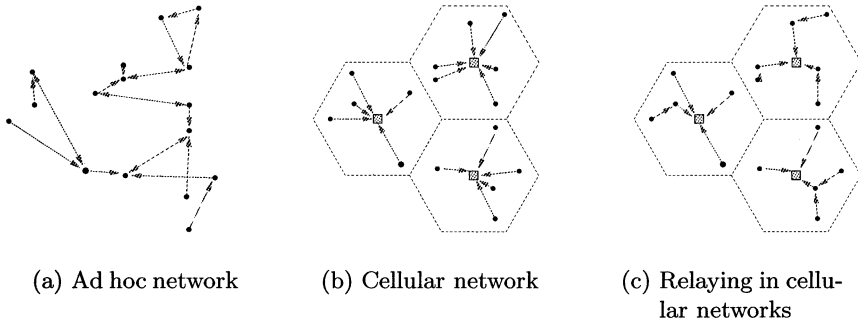


Fig. 1. Mixing communication paradigms

1.2 The Potential of Relaying

For special network arrangements, it has been shown that using relaying can increase the data throughput in particular cases by up to 60% [3]. The method presented there to analyze the achievable data throughput gain is to find best-possible transmission schedules a) with and b) without relaying and compare the resulting data throughput. However, the *average* achievable additional data throughput by allowing relaying as well as a method to solve the scheduling and resource allocation problem for general networks are yet unknown. This work develops linear and mixed integer programming models for the problem as well as adapted algorithms and uses these tools to assess the *average* potential of relaying in *arbitrary* networks. Some work on scheduling and routing in ad hoc networks uses similar models of the problem, see e.g. [1,6,5], a literature survey can be found in [2].

2 Formulating the Optimization Problem

We now outline the scheduling and resource allocation problem we solve to assess the potential of relaying in cellular networks. The details of our model and its computational parameters are based on HiperLAN/2 radio technology [4]. The results, however, apply essentially to any other modern WLAN technology.

2.1 Transmission Schedules and Data Throughput

Our objective is to maximize data throughput in the *uplink*, that is, we want to transmit as much data to the base station as possible in a fixed period of time of T unit time slots. (We assume that each mobile has an unlimited amount of data to send to its base station.) Data throughput is assigned to the mobiles by specifying a global *transmission schedule* for the network.

A transmission schedule consists of a set of *transmission commands*, that specify all parameters for transmitting a single, atomic *data unit*. For each transmission we have to choose the sender and receiver, the transmission power, and the transmission speed. The last parameter can actually not be chosen freely, but there is a fixed set M of physical modes (six in Hiper-LAN/2) to pick from. The number of time slots needed to transmit one data unit varies for the different physical modes, we will denote it by $l_m \in \mathbb{N}$ for a mode $m \in M$. The transmission power has to be picked from a (hardware dependent) interval $[P_{\min}, P_{\max}]$.

An important limitation to the feasibility of transmission schedules is posed by interference. The amount of interference is normally measured relative to the strength of the desired signal (Signal-to-Interference-and-Noise Ratio, SINR). We simplify the relation between SINR and transmission success and assume that a transmission is successful if and only if the SINR stays above a certain threshold ξ , that is,

$$\frac{\text{Received Signal}}{\sum \text{Interfering Signals} + \nu} \geq \xi_m. \quad (\text{SINR})$$

The term ν here stands for an omnipresent amount of thermal noise. The threshold ξ_m varies, it is higher for faster physical modes – higher transmission speed comes at the cost of increased error liability.

2.2 Fairness

If we only strive to maximize the amount of data that is transmitted to the base station, the resulting schedules will be quite unbalanced: stations that are close to the base station will get all the data throughput (and we would never observe relaying). As this behavior is unwanted, we need to add fairness conditions to our problem that ensure that each terminal gets a share of the overall throughput. The actual modeling of fairness is crucial; different fairness models will lead to quite different results. We have implemented two different notions that form the two extremes of the spectrum in which fairness can be conceived:

Time Share Fairness. No terminal should get more throughput than it would get if the available time was split equally between all mobiles and transmission conditions were optimal (that is, it could use the fastest physical mode). On the other hand, no user should get less throughput than it would get under fair time share if it had to use the slowest physical mode.

Total User Fairness. All stations get the same throughput (and this common throughput is to be maximized). This corresponds roughly to maximizing the minimum throughput that any station gets.

3 Mathematical Models

We now describe mathematical models for the relaying problem. We start with a mixed integer model of feasible transmission conditions at a single instance in time; we then develop two different complete models that rely on this basic model and incorporate the time component. The structure and relation between our models is outlined in Fig. 2.

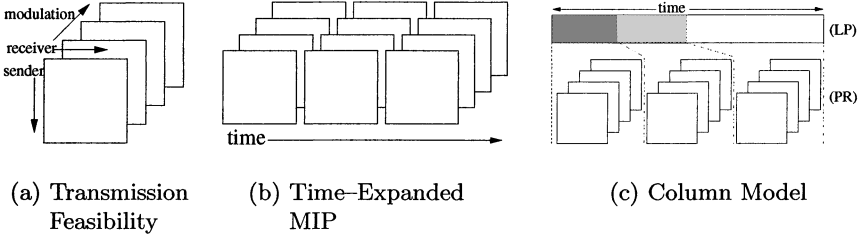


Fig. 2. Structure of Models

3.1 Feasible Transmission Conditions: a Basic Mixed Integer Model

All possible transmissions are denoted by binary variables $x \in \{0, 1\}$, and the associated transmission power is specified in related variables p . If $x = 1$, this means that the referring transmission takes place, otherwise it is not scheduled. We denote the set of all stations by S and the set of available physical modes by M . The index set of all possible transmissions – sketched in Fig. 2(a) – is

$$I := M \times S \times S.$$

We have as variables

$$\begin{aligned} p_{mij} &\in [0, P_{\max}] \quad \forall (m, i, j) \in I, \\ x_{mij} &\in \{0, 1\} \quad \forall (m, i, j) \in I. \end{aligned} \quad (\text{B.1})$$

For any assignment of these variables to form a feasible transmission schedule, we first need to ensure some simple, combinatorial conditions. Any station might only send *or* receive at the same time:

$$\sum_{j \in S} \sum_{m \in M} (x_{mij} + x_{mji}) \leq 1 \quad \forall i \in S. \quad (\text{B.2})$$

Furthermore, the semantics and relation between x and p variables have to be guaranteed, that is,

$$P_{\min} x_{mij} \leq p_{mij} \leq P_{\max} x_{mij} \quad \forall (m, i, j) \in I. \quad (\text{B.3})$$

The most complex set of constraint enforces the interference conditions (SINR). Since these conditions must only be active for actually scheduled transmissions ($x = 1$), we add a “switching” term containing a bound SG_∞ such that the constraint is only sharp if the referring transmission is scheduled. Writing γ_{ij} for the channel attenuation of transmission power between two stations $i, j \in S$ we obtain

$$\gamma_{ij}p_{mij} + \xi_m \text{SG}_\infty(1 - x_{mij}) \geq \xi_m \left(\nu + \sum_{\substack{(n,s,r) \\ \neq (m,i,j)}} \gamma_{si}p_{nsr} \right) \quad \forall (m,i,j) \in I. \quad (\text{B.4})$$

3.2 Time-Expanded Mixed-Integer Model

When compiling a transmission schedule, we have to ensure that feasible transmission conditions prevail at all times. A simple way to do so is to copy the basic model for each time slot. We get variables x_{mijt} and p_{mijt} , where

$$t \in \{1, \dots, T\}$$

denotes the time slot, and we understand $x_{mijt} = 1$ as “Station i begins to transmit a data unit to station j at time t , using modulation m .” Transmission will then be finished at time $t + l_m$. The construction is depicted in Fig. 2(b).

We furthermore take T copies of the constraints and substitute the term

$$\sum_{t \leq s < t+l(m)} x_{mij s}$$

for all occurrences of x_{mij} in the basic equations (B.2)–(B.4) at each time slot t . This provides the necessary link between overlapping transmissions. The time-expanded model including the formulation of the respective objective functions is presented in full detail in [2]. However, computational experiments showed that the model dimensions make the problem intractable for any nontrivial problem instances.

3.3 Column-Based Model

We now present a more sophisticated general model that uses the basic model for the pricing subproblem in a column generation scheme. The results, however, are approximations of actual transmission schedules and have to be rounded. The structure of this model is shown in Fig. 2(c).

Combining Transmission Patterns

The model we use has two tiers. The first tier, formulated as a linear program (LP), is responsible for finding a good distribution of the available time to all possible *transmission patterns*. A transmission pattern is a feasible set of concurrent transmissions and corresponds to a feasible solution $(\mathbf{x}^*, \mathbf{p}^*)$ of our basic model (B). The data throughput that each Station $i \in S$ achieves per time unit for any given transmission pattern is summarized in a matrix Φ . The column corresponding to $(\mathbf{x}^*, \mathbf{p}^*)$ is constructed by letting

$$(\Phi)_{i,*} := \sum_{j \in S} \sum_{m \in M} (x_{mij}^* - x_{mji}^*) / l_m \quad \forall i \in S. \quad (1)$$

Due to the constraints (B.2), at most one of the variables x in the sum is different from zero. The result $(\Phi)_{i,*}$ measures how much data station i puts on the network or receives from another station (negative data throughput) per time unit under a specific transmission pattern.

We write the problem of finding a time weighting $\boldsymbol{\eta}$ of all possible transmission patterns as a linear program. The matrix Φ is used to calculate the net data throughput that any station achieves. We sketch the model for the second fairness model (total user fairness), a full explanation and the model for the alternate fairness concept can be found in [2]. After adding an auxiliary variable $\bar{\alpha}$ to count the minimum data throughput our optimization problem reads

$$\begin{aligned} \max \quad & \bar{\alpha} \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{\eta} \leq T \\ & \Phi \boldsymbol{\eta} \geq \bar{\alpha} \\ & \boldsymbol{\eta} \geq 0 \end{aligned} \quad (\text{LP})$$

Since we can only find a fractional time weighting $\boldsymbol{\eta}$ within this model, the resulting transmission schedule does not take into account that data units cannot be split in reality. A simple rounding-off procedure to cast the result to feasible transmission schedules with tolerable rounding loss is described in [2].

The Pricing Problem

The matrix Φ in the model (LP) has as many columns as there are feasible assignment of \mathbf{x} variables in (B). These are too many to be considered explicitly, we therefore use a *column generation* approach. We start with a reduced set of columns, solve (LP), and calculate the dual variables $(\mathbf{y}^*, \mathbf{w}^*)$ (where \mathbf{y} corresponds to the time constraint and \mathbf{w} to the constraints $\Phi \boldsymbol{\eta} \geq \bar{\alpha}$). We use these dual variables to determine whether there are any columns that are not considered yet but could improve the objective of (LP). This problem is commonly called the *pricing problem* within the framework of column generation. This pricing problem, that is, the search for helpful transmission patterns, constitutes the second tier of our model.

By writing down the dual problem of (LP) we see that (y^*, w^*) is feasible – which in turn means that the set of considered columns is sufficient – if and only if

$$\max_{\substack{\mathbf{x} \text{ feasible in (B)} \\ \text{with suitable } \mathbf{p}}} \sum_{(m,i,j)} \left(\frac{w_i^* - w_j^*}{l_m} \right) x_{mij} \leq y^*. \quad (\text{PR})$$

We can thus solve the pricing problem by adding the left hand side of (PR) as the objective function to (B) and comparing the optimal value to y^* . If the inequality holds true, we are done. Otherwise we can use the vector \mathbf{x} to construct a new column for (LP) as in (1).

4 Results

We implemented a column generation algorithm based on the model described in the previous section that computed almost optimal relaying schedules for special network configurations within a few hours on a standard PC (with an approximation of a few percent). The details can be found in [2]. The main computational work was to solve the pricing problem (PR) at late stages.

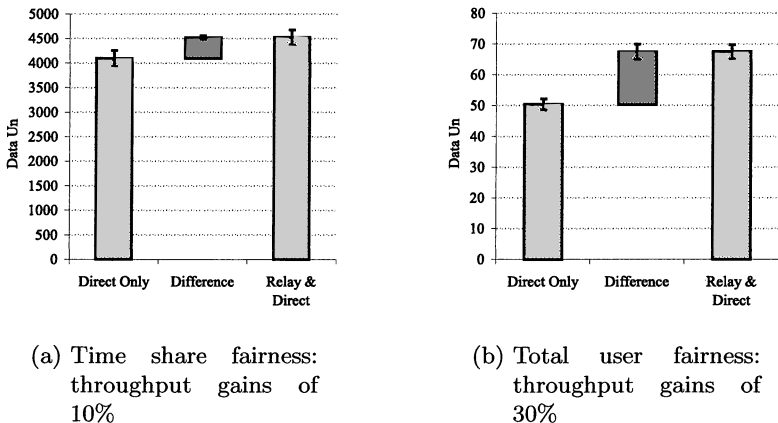


Fig. 3. Computational results

For selected classes of networks this implementation was used to quantify the average throughput gains by relaying. This was carried out by statistically analyzing the results for many “snapshots” of concrete configurations of mobile hosts. For the first fairness model, fair time share, the result can be seen in Fig. 3(a); the average throughput gain of about 10% is rather small. This is because this fairness model favors mobiles that are close to the base station, and these mobiles are unlikely to use relaying. In the total user

fairness model, on the other hand, the gain amounts to about 30%. (All error bars represent 5% confidence intervals.)

Conclusions

We have presented mathematical programming models to solve a complex routing and scheduling problem for wireless cellular networks. These models were used to exemplarily show that the use of relaying can significantly improve data throughput in wireless LANs. This result represents a strong incentive to further develop the relaying approach on the technical side. Further work could, among others, focus on distributed algorithms or investigation of the influence of the fairness concept.

Acknowledgments

This work is based on a master's thesis written under the supervision of Dr. Holger Karl (Technical University Berlin, Telecommunications Network Group) and Dr. Andreas Eisenblätter and Dr. Arie Koster (Zuse Institute Berlin), to all of whom I owe thanks for extensive support. I also thank Prof. Dr. Martin Grötschel and Prof. Dr. Ing. Adam Wolisz.

References

1. P. Björklund, P. Värbrand, and D. Yuan. Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach. In *Proc. IEEE INFOCOM*, San Francisco, CA, March 2003.
2. Hans-Florian Geerdes. Capacity improvements in TDMA-based cellular networks by relaying and flexible transmission scheduling. Master's thesis, TU Berlin, 2002. Available online at http://www.zib.de/groetschel/students/thesis_relaying.ps.zip.
3. H. Karl and S. Mengesha. Analysing capacity improvements in wireless networks by relaying. In *Proc. of IEEE Intl. Conf. on Wireless LANs and Home Networks*, pages 339–348, Singapore, December 2001.
4. J. Khun-Jush, G. Malmgren, P. Schramm, and J. Torsner. HIPERLAN type 2 for broadband wireless communication. *Ericsson Review*, 2:108–119, 2000.
5. S. Kim, Z. Rosberg, and J. Zander. Combined power control and transmission rate selection in cellular networks. In *Proc. of the 49th IEEE Vehicular Technology Conference Fall*, pages 1653–1657, Amsterdam, The Netherlands, September 1999.
6. Stavros Toumpis and Andrea Goldsmith. Capacity regions for wireless ad hoc networks. submitted, 2002.

Performance Analysis of M -designed Inbound Call Centers

Raik Stolletz

University of Hannover, Department of Production Management, Königsworther Platz 1, 30167 Hannover, raik.stolletz@prod.uni-hannover.de

Abstract. Many call centers provide service for customers of different classes. We analyze a queueing model of an inbound call center with two customer classes, three agent groups, and skills-based routing. In our model we assume that a waiting customer may hang up before his service begins. All times are assumed to be exponentially distributed. We describe the states and the state space of this continuous time Markov chain and develop the steady-state equations. The behavior of this system is analyzed in numerical experiments and optimal economical allocations of the agents are discussed.

1 Introduction

Call centers provide phone-based services to customers or clients in the private or public sector. During the past years, the number of call centers and the number of agents has grown rapidly, demonstrating the increasing importance of call centers. By the direction in which the contact between the customer and the agent is initiated, we distinguish inbound and outbound call centers. In *inbound call centers* the employees or agents receive calls from outside customers, and therefore these call centers are driven by random customer call arrivals.

The performance of inbound call centers can be measured by technical performance measures such as waiting times, availability of service, or customer abandonment. These technical performance measures improve if the call center management employs more agents. However, the operating costs of call centers are mainly driven by the costs of these agents. About 60 to 70 percent of the operating costs are personnel-related. Call center management has to adjust imbalances of acceptable technical performance measures and the economic performance of the call center. Both the number of agents and the number of offered phone lines are important decision variables in this context.

Research in different scientific disciplines is related to call centers, for example, studies in Operations Research, Marketing, or Information Technology, see the comprehensive research bibliography of call center-related literature of Mandelbaum [8]. An overview of formal models and operational planning problems in call centers is given by Gans et al. [2], Helber and Stolletz [4,5], and Stolletz [11].

Many inbound call centers provide service for heterogeneous customers with heterogeneous agents. In such call centers, the customers can be routed to different agent groups and the agents can serve customers of different classes, which is referred to as *skills-based routing*. As an example of skills-based routing, this paper analyzes a queueing model of an inbound call center with heterogeneous customers and heterogeneous agents, the so-called *M*-design. In Section 2, we describe this queueing model with the considered routing policies. A summary of the derivation of steady-state probabilities and technical as well as economic performance measures is given in Section 3. Section 4 presents some numerical results.

2 Description of the Queueing Model

This section describes a model of an inbound call center with two classes (A and B) of customers and three different groups of agents, as depicted in Figure 1. This queueing system is completely described by customer profiles, agent characteristics, routing policies, and the limitation of the waiting room.

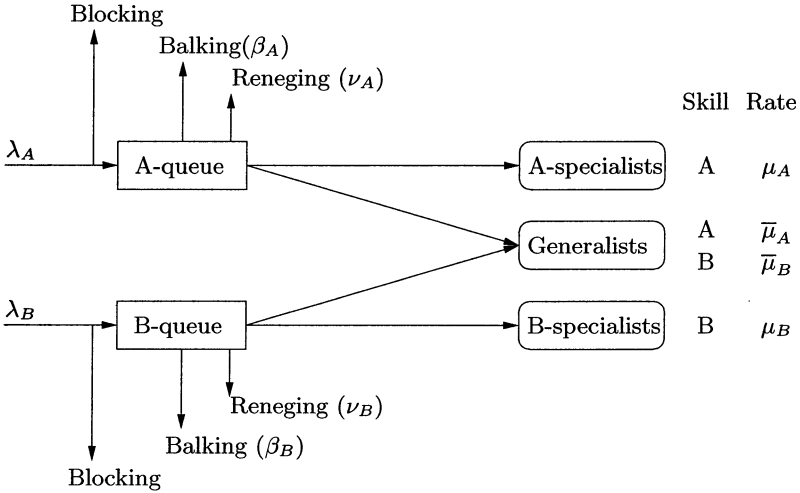


Fig. 1. Schematic model of an *M*-design call center

Customer Profiles: A-customers and B-customers arrive at the system according to independent homogeneous Poisson processes with arrival rates λ_A and λ_B , respectively. It is assumed that customers of both classes are impatient. If an arriving A-customer cannot reach an agent immediately, he balks with probability β_A or joins the queue with probability $1 - \beta_A$. A B-customer balks with probability β_B or joins the queue with probability $1 - \beta_B$.

A waiting customer of any class reneges after an exponentially distributed waiting time limit, if the service has not begun. Let ν_A and ν_B be the rates of these distributions of the waiting time limits for A- and B-customers, respectively. Abandoned customers are assumed to be lost, i.e., there are no retrials.

Agent Characteristics: The three groups of agents have different skills. Each agent belongs to just one group and works with exponentially distributed processing times. The c_A agents of the first group are specialists for A-customers and each agent works with an average processing rate μ_A . The c_G flexible agents of the second group are generalists crosstrained for A- as well as for B-customers. Each generalist works with an average processing rate $\bar{\mu}_A$ for A-customers and $\bar{\mu}_B$ for B-customers, respectively. The c_B agents of the third group are specialists for B-customers and each agent of this group works with an average processing rate μ_B .

Routing Policies: There are two kinds of routing policies describing the service system. The agent selection policy describes how a calling customer is routed. The question of which waiting customer is served next if an agent becomes available is answered by the customer selection policy. For *agent selection*, we assume a preferred-agent-group policy for both customer classes. If a specialist is available, an arriving customer will be served immediately by him. Otherwise, if all specialists are busy, but a generalist is available, this flexible agent serves the arriving customer. If all specialists for this class of customers and all generalists are busy, the arriving customer joins the queue or balks. For *customer selection*, we have different rules for specialists and generalists. The specialists serve customers according to the First-Come-First-Served rule within their particular class. If a generalist becomes available, he serves a waiting customer according to a priority rule. The generalist looks at the A-queue first. If an A-customer is waiting, the generalist serves an A-customer according to the First-Come-First-Served rule. Otherwise, the generalist looks at the B-queue and serves a B-customer according to the First-Come-First-Served rule. If there is no customer in queue, the generalist idles. Therefore, the routing policies follow a non-preemptive priority rule for high-priority A-customers.

Limitation of the Waiting Rooms: Each class of customers has its own queue. We assume that the maximum number K_A of A-customers (waiting and in service) is finite. The maximal number K_B of B-customers in the system is finite as well. This limitation of the system size with completely partitioned waiting rooms is motivated by a given number of telephone trunks for each class of customers.

This model, with all the described features, is analyzed in [11]. In the literature, queueing models of an M -design are analyzed with patient customers only. An M -design without any waiting positions is considered by Sato and Mori [10], Kawashima [7], and Pinker and Shumsky [9]. A loss-waiting sys-

tem is analyzed by Gurumurthi and Benjaafar [3], but they assume that the generalists serve both customer classes with the same mean processing time.

3 Derivation of Steady-State Performance Measures

The stochastic process of the considered call center model is a Markov process, i.e., future behavior depends only on the present, not the past. We analyze the model in steady-state where the probabilities of the different system states do not change over time. A representation of the states of this queueing system must contain information about the number of waiting customers of each class, the number of busy specialists of each class, the number of generalists serving A-customers, and the number of generalists serving B-customers. We describe the states of the system using a quadruple (i, j, k, l) , where

- i is the number of A-customers in the system (waiting and in service),
- j is the number of B-customers in the system (waiting and in service),
- k is the number of generalists serving A-customers, and
- l is the number of generalists serving B-customers.

The number of busy specialists and the number of waiting customers for a given state can be derived from the corresponding tuple (i, j, k, l) as

- $\min\{i - k, c_A\}$ for the number of busy A-specialists,
- $\min\{j - l, c_B\}$ for the number of busy B-specialists,
- $\max\{i - k - c_A, 0\}$ for the number of waiting A-customers, and
- $\max\{j - l - c_B, 0\}$ for the number of waiting B-customers.

Therefore, the quadruple (i, j, k, l) describes the state completely. Steady-state equations of states with waiting customers have a common basic structure, but differ structurally from those equations for states without waiting customers. Therefore, we divide the state space \mathcal{S} into the following four regions:

1. *Region I*: States $(i, j, k, l) \in \mathcal{S}$ with waiting customers of both classes
2. *Region II*: States $(i, j, k, l) \in \mathcal{S}$ with waiting A-customers and without waiting B-customers
3. *Region III*: States $(i, j, k, l) \in \mathcal{S}$ with waiting B-customers and without waiting A-customers
4. *Region IV*: States $(i, j, k, l) \in \mathcal{S}$ without waiting customers

Let $\mathbf{p}(i, j, k, l)$ be the steady-state probability to be in state $(i, j, k, l) \in \mathcal{S}$. The derivation of general steady-state equations for each region is given in [11]. To solve the resulting set of all steady-state equations, we apply the power method. This iterative approach can be applied for discrete-time Markov chains, see Bolch et al. [1] pp. 132-136. To use the power method for continuous-time Markov chains, we transform the basic problem via uniformization, see [1] pp. 103-104. The continuous-time Markov chain which

we analyze is irreducible and finite. Therefore, the Markov chain is ergodic and the power method converges.

Within the resulting steady-state probabilities, some technical performance measures can be derived for both customer classes. The expected number of waiting customers is the sum of the queue lengths $L(z)$ weighted by their steady-state probabilities $\mathbf{p}(z)$ over all states $z \in \mathcal{S}$. To derive the expected length $\mathbf{E}[L_A]$ of the A-queue, the sum goes over all states with waiting customers of both classes (Region I) and states with waiting A-customers and no waiting B-customers (Region II), i.e.,

$$\mathbf{E}[L_A] = \sum_{k=0}^{c_G} \sum_{i=c_A+k+1}^{K_A} \sum_{j=c_G-k}^{K_B} (i-k-c_A) \mathbf{p}(i, j, k, c_G-k). \quad (1)$$

A calling A-customer receives service immediately if there is at least one available A-specialist or generalist. Therefore, the probability of receiving service immediately is

$$\begin{aligned} P(W_A = 0) &= \sum_{(i,j,k,l) \in \text{Region III: } i \neq c_A+k} \mathbf{p}(i, j, k, l) \\ &+ \sum_{(i,j,k,l) \in \text{Region IV: } i \neq c_A+k \text{ or } c_G \neq k+l} \mathbf{p}(i, j, k, l). \end{aligned} \quad (2)$$

Other performance measures can be derived similarly for A- as well as for B-customers. Besides these performance measures for each customer class, we analyze *weighted performance measures* for both customer classes together. A measure for a particular class is weighted by the fraction of calling customers in this class. For example, the weighted probability of receiving service immediately $P_{A+B}(W = 0)$ is given by

$$P_{A+B}(W = 0) = \frac{\lambda_A P(W_A = 0) + \lambda_B P(W_B = 0)}{\lambda_A + \lambda_B}. \quad (3)$$

Based on these technical performance measures, we can derive economic performance measures of the call center. Dependent on the used service numbers we distinguish three classes of services: toll-free, shared-cost, and value-added services, see Helber et al. [5,6]. If we consider a sales call center providing service via toll-free or shared-cost numbers, the cost function consists of telephone costs and agent costs. Let \mathcal{C}^{u_A} and \mathcal{C}^{u_B} be the telephone cost per time unit for an A- and a B-customer on hold, respectively. For the cost of an agent per time unit we use the notation \mathcal{C}^A , \mathcal{C}^B , and \mathcal{C}^G for agents of the

particular group. Then we have the cost function

$$\begin{aligned}
\mathcal{C}(c_A, c_B, c_G, K_A, K_B) = & \underbrace{(\mathbf{E}[L_A] + c_A \mathbf{E}[u_A] + c_G \mathbf{E}[u_{G_A}])}_{\text{telephone costs for A-customers}} \mathcal{C}^{u_A} \\
& + \underbrace{(\mathbf{E}[L_B] + c_B \mathbf{E}[u_B] + c_G \mathbf{E}[u_{G_B}])}_{\text{telephone costs for B-customers}} \mathcal{C}^{u_B} \\
& + \underbrace{\mathcal{C}^A c_A + \mathcal{C}^B c_B + \mathcal{C}^G c_G}_{\text{agent costs}}, \tag{4}
\end{aligned}$$

where $\mathbf{E}[u_A]$ and $\mathbf{E}[u_B]$ are the expected utilizations of the specialist groups. $\mathbf{E}[u_{G_A}]$ and $\mathbf{E}[u_{G_B}]$ are the expected utilizations of the generalists with A- and B-customers, respectively.

A call center can generate revenue per served customer, for example in a sales call center. We consider the case, where the expected revenue \mathcal{R}_A^S and \mathcal{R}_B^S per served customer is independent of the talk time. We derive the expected revenue per time unit via the probabilities of receiving service, i.e.

$$\mathcal{R}^S(c_A, c_B, c_G, K_A, K_B) = \lambda_A P_A(\text{ service }) \mathcal{R}_A^S + \lambda_B P_B(\text{ service }) \mathcal{R}_B^S. \tag{5}$$

The difference of the revenue (5) and the costs (4) gives the profit \mathcal{P} .

4 Numerical Results

In this example, we show how the allocation of a fixed number of agents into the three agent groups influences technical as well as economic performance measures. Consider a call center with $c = c_A + c_B + c_G = 20$ agents. The arrival rates are assumed to equal $\lambda_A = \lambda_B = 300$ calls per hour. We assume that waiting customers of both classes renege after an average waiting time of 20 seconds, i.e., $\nu_A^{-1} = \nu_B^{-1} = 20$ seconds. Specialists and generalists may have different skill levels such that the average processing times differ. The generalists are trained to handle both types of customers. This might result in greater processing times for the generalists. We assume that the specialists of both groups have an average processing time of $\mu_A^{-1} = \mu_B^{-1} = 100$ seconds, and a generalist needs on average 20 seconds more to handle a call, i.e., $\bar{\mu}_A^{-1} = \bar{\mu}_B^{-1} = 120$ seconds. We assume that the numbers of telephone trunks are $K_A = K_B = 30$.

Now, we vary the number of agents c_A , c_B , and c_G such that each group is staffed with at least one agent, and all 20 agents belong to one of the agent groups, i.e., $c = c_A + c_B + c_G = 20$. Figure 2 depicts the weighted probability $P_{A+B}(W = 0)$ of receiving service immediately. This probability has a maximum of $P_{A+B}(W = 0) = 78.23\%$ for the allocation $c_A = c_B = 8$ and $c_G = 4$. If the call center is staffed very asymmetrically, i.e., we have many specialists in only one agent group, the probability $P_{A+B}(W = 0)$ is relatively low.

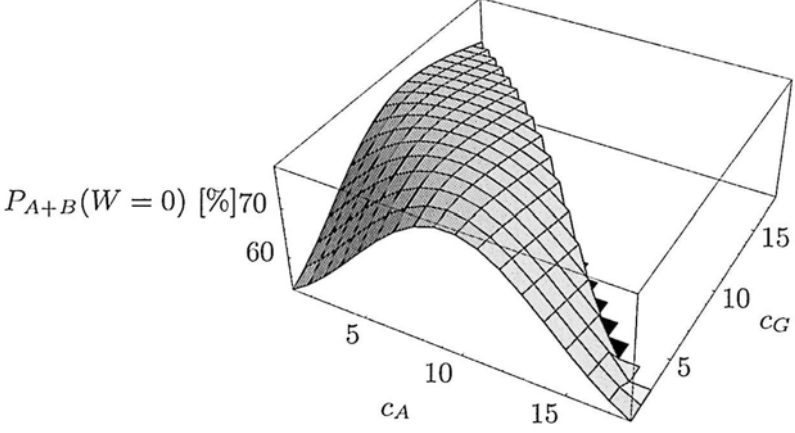


Fig. 2. Weighted probability of receiving service immediately for allocations of $c = 20$ agents

Table 1. Optimal allocation of agents, such that each agent group is staffed with at least one agent and $c = c_A + c_B + c_G = 20$

$\mu_A^{-1} = \mu_B^{-1}$ [sec.]	c_A^*	c_B^*	c_G^*	$\mathcal{P}(c_A^*, c_B^*, c_G^*)$ [euros /min.]	$\frac{c_G^*}{c_A^* + c_B^* + c_G^*}$ [%]	$P_{A+B}(\text{service})$ [%]
120	1	1	18	78.52	90	87.34
115	3	3	14	78.82	70	87.61
110	6	6	8	79.79	40	88.51
105	7	7	6	81.13	30	89.78
100	7	7	6	82.59	30	91.21
95	8	8	4	84.12	20	92.64
90	8	8	4	85.60	20	94.06

In Table 1 we analyze how the economic performance of such a call center depends on the processing times of the agents. We consider a profit function \mathcal{P} of a sales call center described in Section 3. We assume telephone costs of $\mathcal{C}^{u_A} = \mathcal{C}^{u_B} = 0.12$ euros per minute and agent costs of $\mathcal{C}^A = \mathcal{C}^B = \mathcal{C}^G = 20$ euros per hour. A served customer generates an expected revenue of $\mathcal{R}^S = 10$ euros, independent of the customer class. Table 1 gives the economically optimal allocation of $c = 20$ agents for different mean processing times of the specialists and impatient customers ($\nu_A^{-1} = \nu_B^{-1} = 20$ seconds). A generalist provides service with an average processing time of $\bar{\mu}_A^{-1} = \bar{\mu}_B^{-1} = 120$ seconds. If the mean processing time of the specialists decreases, the percentage of generalists in the optimal allocation decreases as well. Hence, the effects of pooling decrease, and the influence of advantages of shorter processing times on the profit function increases. The last column of Table 1 gives the weighted

probability P_{A+B} (service) of receiving service. If the average processing time of the specialists decreases, more customers of both classes receive service.

Although a generalist needs more time to serve a customer than a specialist, the optimal allocations have more than one generalist. This holds with respect to a single performance measure as well as for the profit function of a sales call center. However, for an increasing difference between the mean processing times of specialists and generalists, the advantages of pooling agents into a group of generalists decrease.

References

1. Bolch, G., Greiner, S., de Meer, H. and Trivedi, K. S. (1998). *Queueing networks and Markov chains: Modeling and performance evaluation with computer science applications*. John Wiley & Sons, Inc., New York et al.
2. Gans, N. and Koole, G. and Mandelbaum, A. (2003). Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing & Service Operations Management*, 5(2):79-141.
3. Gurumurthi, S. and Benjaafar, S. (2001). *Modeling and Analysis of Flexible Queueing Systems*. Department of Mechanical Engineering, Division of Industrial Engineering, University of Minnesota.
4. Helber, S. and Stolletz, R. (2001). Grundlagen der Personalbedarfsermittlung in Inbound-Call Centern. To appear in: *Zeitschrift für Betriebswirtschaft*.
5. Helber, S. and Stolletz, R. (2003). *Call Center Management in der Praxis. Strukturen und Prozesse betriebswirtschaftlich optimieren*. Springer-Verlag, Berlin et al.
6. Helber, S., Stolletz, R. and Bothe, S. (2002). Erfolgszielorientierte Agentenallokation in Inbound Call Centern. To appear in: *Zeitschrift für betriebswirtschaftliche Forschung*.
7. Kawashima, K. (1985). An approximation of a loss system with two heterogeneous types of calls. *Journal of the Operations Research Society of Japan*, 28(2):163-177.
8. Mandelbaum, A. (2003). *Call Centers (Centres). Research Bibliography with Abstracts*. Industrial Engineering and Management, Technion, Haifa 32000, Israel. URL: <http://ie.technion.ac.il/serveng>.
9. Pinker, E. J. and Shumsky, R. A. (2000). The Efficiency-Quality Trade-Off of Cross-Trained Workers. *Manufacturing & Service Operations Management*, 2(1):32-48.
10. Sato, T. and Mori, M. (1983). An Application of the Lumping Method to a Loss System with Two Types of Customers. *Journal of the Operations Research Society of Japan*, 26(1):51-59.
11. Stolletz, R. (2003). *Performance Analysis and Optimization of Inbound Call Centers*. Lecture Notes in Economics and Mathematical Systems, Vol. 528. Springer-Verlag, Berlin et al.

Revenue Management in Manufacturing

Florian Defregger and Heinrich Kuhn

Catholic University of Eichstätt-Ingolstadt, 85049 Ingolstadt, Germany

Abstract. Revenue Management has proven successful in a number of service industries such as airlines, hotels and cruiseships. This paper examines the possibility of applying revenue management techniques to a manufacturing setting. A make-to-order company is considered which receives orders with different profit margins, processing times and due dates. The corresponding discrete markov decision process model is presented as well as a heuristic which solves large problem instances within a reasonable runtime.

1 Introduction

Revenue Management originates in service industries like the airline and hotel businesses. Here we explore the possibility of applying revenue management techniques to a manufacturing company [1]. We consider a make-to-order company with high fixed costs. Figure 1 shows an example of the decision process.

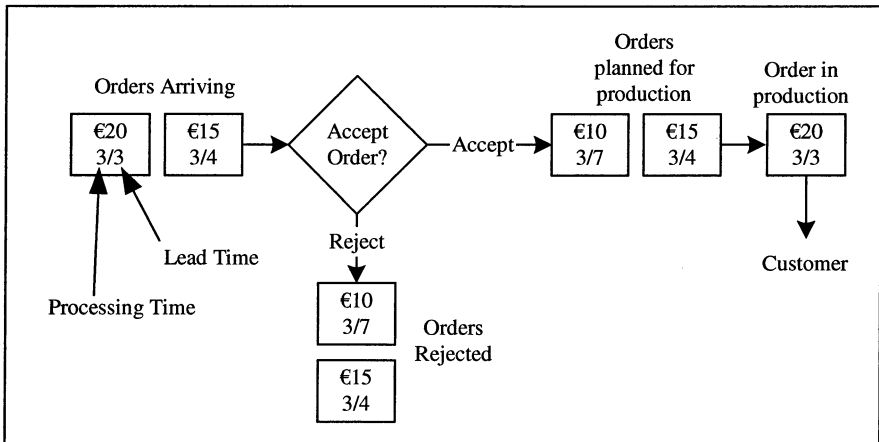


Fig. 1. Order Selection Process

Stochastic orders, each with its own profit margin, lead time and processing time arrive at the company which has to decide which orders to accept and which ones to refuse. Accepting too many orders with low profit margins

might clog the production capacity so much that orders with higher profit margins will have to be turned down once they arrive. In order to avoid this, some orders with low profit margins should be rejected even if capacity to accept them is available. But if too many of those orders are rejected the company might lose money as well. Thus the company has to decide in which situations orders with lower profit margins should be rejected.

2 Model

Kniker and Burman [2] present an infinite-horizon markov decision process model which is characterized as follows. All orders arriving at the company belong to a certain order class $n \in \{1, \dots, N\}$. In each discrete time period at most one order can arrive with probability p_n , $\sum_{n=1}^N p_n < 1$. All orders of class n have a profit margin of m_n monetary units, a processing time of r_n a lead time of l_n time periods. The lead time specifies that the customer is willing to wait for a maximum of l_n time periods when placing the order at the company. To model the case that no order arrives in a given period, the dummy order class 0 is used, $p_0 = 1 - \sum_{n=1}^N p_n$, $m_0 = r_0 = l_0 = 0$.

Each system state is characterized by (n, c) , with n being the order class which has arrived at the beginning of the current period and c being the number of periods the capacity is booked out by orders which have already been accepted and not been completed yet, $c \in \{0, \dots, \max_{n \in \{1, \dots, N\}} l_n\}$. The set of all system states is denoted by S , $|S| = (N + 1) \cdot \max_{n \in \{1, \dots, N\}} l_n$.

The state-dependent actions A of the markov decision process are given by

$$A[(n, c)] = \begin{cases} A1 := \text{"reject"} \\ A2 := \text{"accept"} : n > 0 \wedge c + r_n \leq l_n \end{cases}$$

Orders can always be rejected, in the case no order arrives $A1$ has the meaning of "wait for the next order to arrive". An order can only be accepted if the order can be finished within the lead time l_n .

Action- and state-dependent rewards R are given by

$$\begin{aligned} R^{A1}[(n, c)] &= 0, & \forall (n, c) \in S \\ R^{A2}[(n, c)] &= m_n, & \forall (n, c) \in S \end{aligned}$$

The company receives a reward of m_n if it accepts an order of class n , otherwise there is no reward.

The transition probabilities in the case of rejecting an order are:

$$P^{A1}[(n, c), (m, c - 1)] = \begin{cases} p_m, & \forall (n, c) \in \{S : c \neq 0\}, m \in \{0, \dots, N\} \\ 0, & \text{else} \end{cases}$$

$$P^{A1}[(n, 0), (m, 0)] = \begin{cases} p_m, & \forall n, m \in \{0, \dots, N\} \\ 0, & \text{else} \end{cases}$$

By rejecting an order or waiting for the next order to arrive the capacity usage c is decreased by 1. If $c = 0$, c stays 0.

The transition probabilities for accepting an order are:

$$P^{A2}[(n, c), (m, c + r_n - 1)] = \begin{cases} p_m, & \forall (n, c) \in S, m \in \{0, \dots, N\} \\ 0, & \text{else} \end{cases}$$

When accepting an order, c is increased by r_n , but will be decreased by 1 in the next period because of the ongoing production.

By solving the markov decision process an optimality criterion is optimized by finding the optimal policy π which maps an optimal action to each state (n, c) .

The optimality criterion chosen is the average reward per period which can be maximized by standard methods like policy iteration or value iteration. As the runtime of these algorithms increases quickly with larger $|S|$, a heuristic is needed for large problem instances.

3 Heuristic

Let $n \in \{1, \dots, N\}$ be the order classes sorted ascending by their relative profit margins m_n/r_n (this can easily be achieved by rearranging the order classes accordingly).

The assumption is made that by gradually rejecting orders of the classes $\{1, \dots, N\}$ the average reward increases monotonously to a global optimum and decreases monotonously after that. This is illustrated for 2 order classes in Figure 2.

Starting out with the FCFS-Policy, which accepts all incoming orders as long as their leadtimes are not violated, orders are gradually rejected in class 1 until reaching a global optimal average reward at $c = 5$. By rejecting even more orders, the average reward decreases.

The average reward curve includes for each order class n all the capacity usages c where orders can be accepted. For each order class, orders can be accepted as long as the lead times are not violated, i.e. $c \leq l_n - r_n$. Thus, for each order class n , c ranges from 0 to $l_n - r_n$.

Each dot is associated with a specific machine usage c and order class n and symbolizes the policy of rejecting orders in states "left to it" and accepting orders in states "right to it" on the graph. Each of these policies has an average reward associated with it.

For example, the policy with the highest average reward at $n = 1$ and $c = 5$ means that the company should reject orders of order class 1 as long

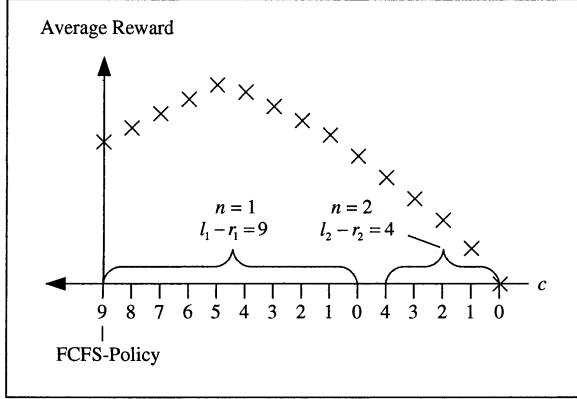


Fig. 2. Assumption about the Average Reward Curve

as the current capacity is booked out for more than $c = 5$ periods and accept orders of class 1 otherwise. All orders of class 2 should be accepted.

The heuristic starts by numbering all policies from 1 to P , with P being the number of policies given by

$$P = \sum_{i=1}^N (l_i - r_i + 1)$$

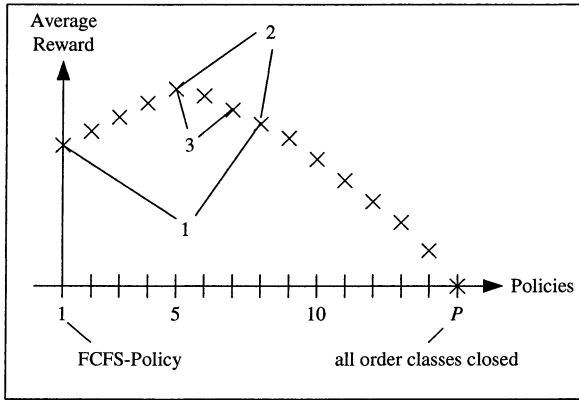


Fig. 3. Sequence of Comparisons

As shown in Figure 3, the policies for the example given are numbered from 1 to 15. After numbering the policies the heuristic tries to find the

assumed global optimal policy by comparing policies pairwise. Each pairwise comparison is done by simulation, following the paired-t confidence interval approach [3]. During each comparison the average rewards of the two policies concerned are compared until a significant difference has been detected or the two average rewards differ by less than 0.5%. Figure 3 illustrates the sequence of pairwise comparisons that the heuristic might typically take.

4 Numerical Results

Numerical results for the heuristic are presented in Table 1. Calculations were carried out on a 233-MHz Pentium machine. The performance of the heuristic was compared to a standard solution procedure for markov decision processes, namely value iteration. The average reward obtained by value iteration deviated at most 0.1% from the optimal average reward of the markov decision process. For 64000 states this maximum deviation was lowered to 1%, 3% and 5% in order to show how the runtime of value iteration would behave with these percentages.

The examples show that the heuristic is able to solve even large problem instances in reasonable runtimes. The heuristic misses optimal solutions at high numbers of states because it limits its search to policies on the average reward curve shown in Figures 2 and 3. Other policies which implement individual booking limits for each order class n are possible, though, and might achieve higher average rewards as is demonstrated for $|S| = 64000$.

Table 1. Numerical Results

$ S $	Average Reward			Runtime [sec.]	
	Value Iteration	Heuristic	% Diff.	Value Iter.	Heuristic
1000	7.617	7.610	0.09%	67	45
2000	7.889	7.891	-0.03%	360	66
4000	2.788	2.760	1%	464	169
8000	1.646	1.646	0.01%	2052	444
16000	0.540	0.535	0.93%	8395	529
32000	1.013	1.010	0.36%	23352	625
64000	0.716	0.688	4.14%	115495	562
64000	0.715 [1%]	0.688	3.93%	76883	562
64000	0.713 [3%]	0.688	3.61%	59364	562
64000	0.712 [5%]	0.688	3.52%	48741	562

References

1. H. Kuhn and F. Defregger. Revenue Management in der Sachgüterproduktion. *WiSt – Wirtschaftswissenschaftliches Studium*, 32(12), 2003.
2. T. S. Kniker and M. H. Burman. Applications of revenue management to manufacturing. In *Third Aegean International Conference on Design and Analysis of Manufacturing Systems, May 19-22, 2001, Tinos Island, Greece*, pages 299–308, Thessaloniki, Greece, 2001. Editions Ziti.
3. A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*, chapter 10. McGraw-Hill, 2000.

Network Revenue Management: Some Issues on Upper and Lower Bounds

Michael Müller-Bungart

TU Bergakademie Freiberg, Lessingstr. 45, 09596 Freiberg, Germany,
michael.mueller-bungart@bwl.tu-freiberg.de

Abstract. We consider a Network Revenue Management Problem with uncertain demand and nested capacity control. The objective is to maximize expected revenue by setting nested booking limits b_j for all products $j = 1, \dots, n$. For this setting, no closed form for the objective function is known. In this paper, we propose a method to derive upper and lower bounds on the expected revenue when nested booking limits b_1, \dots, b_n are given.

1 The Network Revenue Management Problem

Consider the following planning situation:

- Multiple resources are used to produce a variety of products, so that the same unit of resource may yield a different revenue.
- The capacity of these resources is fixed, so that it is (practically) impossible to adjust capacity according to demand.
- Demand is likely to exceed available capacity, so that some requests have to be rejected.

Under these circumstances, rules to decide which requests should be rejected have to be developed. Obviously, a simple myopic acceptance/rejection policy like “first come, first serve” will lead to a suboptimal allocation of resources to demand in revenue terms. So the purpose of Network Revenue Management (NRM) methods is to divide resource capacities among products in a way that maximizes revenue. A common approach to control capacity usage is to define so called “booking limits” b_j for each product $j = 1, \dots, n$ so that at most b_j requests will be accepted for product j .

For a more detailed description of the NRM problem and an extensive literature overview the reader is referred to the surveying papers by Klein [6], McGill and van Ryzin [7], Tscheulin and Lindenmeier [10], and Weatherford and Bodily [11].

2 Problem Setting

2.1 Notation

In general, we can state a NRM Problem as follows: We are given $m \geq 1$ resources with fixed capacities $R_i > 0, i = 1, \dots, m$. The cost to supply these capacities is sunk, and thus not relevant to our decisions.

In the planning horizon, we can use the resources to produce n products. The production coefficient of product $j = 1, \dots, n$ with respect to resource i is given by $r_{ij} \in [0, R_i]$. We assume that each product needs at least one resource and that each resource is used by at least two products. The sale of a unit of product j yields a constant revenue (or profit margin) of $v_j > 0$. Demand for product j is uncertain and denoted by the random variable $D_j \in \mathbb{N}_0$, and $P(D_j = k)$ and $P(D_j \geq k)$ are the probabilities that demand for product j is exactly k or at least $k \in \mathbb{N}_0$, respectively.

To control usage of the scarce resource capacities, we enforce booking limits b_j for each product j , i. e. we will at most sell b_j units of product j . It is obviously reasonable to require

$$b_j \in \{0, \dots, \bar{b}_j\} \text{ where } \bar{b}_j = \min \{ \lfloor R_i / r_{ij} \rfloor \mid i = 1, \dots, m, r_{ij} > 0 \} \quad (1)$$

Using booking limits $b = (b_1, \dots, b_n)$ over the planning horizon yields a total revenue of $V(b)$ (a random variable). We want to set the limits b_j so that the expected revenue $E[V(b)]$ is maximized.

2.2 Nested Capacity Control

Two different interpretations for the booking limits b_j have been proposed: On the one hand, the booking limits b_j may be used to *partition* the available capacity, i. e.

$$\sum_{j=1}^n b_j r_{ij} \leq R_i \quad i = 1, \dots, m \quad (2)$$

holds and $b_j r_{ij}$ units of each resource $i = 1, \dots, m$ are exclusively reserved for product $j = 1, \dots, n$. This capacity control strategy is quite inflexible, because the $(b_j + 1)$ -th request for product j will always be rejected even if there is plenty of capacity remaining. To avoid this disadvantage, contingents of capacity should be *nested* among products: A nesting order, i. e. a permutation π of $\{1, \dots, n\}$ is defined and each unit of capacity that is available to product $\pi(j)$ is also available to all products $\pi(j-1), \dots, \pi(1)$. Consequently, we require

$$b_{\pi(j)} \geq b_{\pi(j+1)} \quad j = 1, \dots, n-1 \quad (3)$$

for nested booking limits b_j . The values

$$p_{\pi(j)} = b_{\pi(j)} - b_{\pi(j+1)} \quad j = 1, \dots, n-1 \quad (4)$$

are called *nested protection levels*. At any time, $p_{\pi(j)} r_{i, \pi(j)}$ units of resource $i = 1, \dots, m$ are protected for product $\pi(j)$ from products $\pi(j+1), \dots, \pi(n)$ (Belobaba [1]).

The increase in flexibility of the nested capacity control strategy comes at a cost: The availability of resources depends on the sequence of requests

and acceptance/rejection decisions made earlier. Therefore a dynamic model necessary to represent the situation; in particular, the expected value function $E[V(b)]$ has to be expressed as a recursion formula. Evaluating this formula by Dynamic Programming (DP) is possible, but impracticable except for small instances.

2.3 Literature Review and Purpose of the Paper

Numerous authors have developed dynamic models for NRM problems with nested capacity control. Unfortunately, an efficient method to compute the exact value of $E[V(b)]$ for instances of moderate size has not been found yet. To overcome this issue, Bertsimas and Popescu [3] and Chen et al. [4] have proposed methods to approximate this function. Bertsimas and de Boer [2] estimate $E[V(b)]$ by simulation. Furthermore, heuristics have been developed, the so called bid price method being the most successful (Cooper [5], Simpson [8], Talluri and van Ryzin [9]).

Both to approximate the expected value function and to evaluate the performance of heuristics bounds on $E[V(b)]$ would be helpful. To the author's knowledge, this topic has not received much attention yet, so this paper derives upper and lower bounds on $E[V(b)]$ for a given vector of booking limits b .

3 Deriving Upper and Lower Bounds on $E[V(b)]$

3.1 Deriving a Formula for $E[V(b)]$

Let $d = (d_1, \dots, d_n)$ be a realization of demand, i. e. the event $D_j = d_j, j = 1, \dots, n$ has occurred. The probability of that event is

$$P(d) = \prod_{j=1}^n P(D_j = d_j) \quad (5)$$

Denote the set of all demand realizations with positive probability by

$$D = \{d = (d_1, \dots, d_n) | P(d) > 0\} \quad (6)$$

Finally, let the revenue obtained given demand d and booking limits b be given by $v_b(d)$. Then, we can write

$$E[V(b)] = \sum_{d \in D} P(d) v_b(d) \quad (7)$$

Since $|D|$ may be infinite, we replace D with

$$D_b = \{d \in D | d_j \leq b_j\} \quad (8)$$

Clearly, $|D_b|$ is finite. Define

$$P'(d_j) = \begin{cases} P(D_j = d_j) & d_j \in \{0, \dots, b_j - 1\} \\ P(D_j \geq b_j) & d_j = b_j \end{cases} \quad (9)$$

$$P'(d) = \prod_{j=1}^n P'(d_j) \quad (10)$$

Obviously,

$$E[V(b)] = \sum_{d \in D_b} P'(d) v_b(d) \quad (11)$$

because if $d_j > b_j$, the last $d_j - b_j$ requests will be denied anyway, no matter when they appear in the demand sequence.

Equation (11) is not too helpful to compute $E[V(b)]$ because a closed formula for $v_b(d)$ is unknown. In the following, we will develop (mixed) integer programs that over- or underestimate $v_b(d)$.

3.2 Overestimating $v_b(d)$

Let booking limits $b = (b_1, \dots, b_n)$ be given. Without loss of generality, we assume that

$$b_1 \geq \dots \geq b_n \quad (12)$$

so that we can omit π .

As mentioned before, demand materializes in some sequence, and for some requests resource capacity is allocated, others are rejected. Denote the number of requests finally accepted for product j by $a_j \in \mathbb{N}_0, j = 1, \dots, n$. We call the corresponding vector $a = (a_1, \dots, a_n) \in \mathbb{N}_0^n$ an *allocation*. a has to satisfy the restrictions imposed by the resource capacity, by the booking limits and by the nested protection levels. Formally, the following constraints have to hold:

$$\sum_{j=1}^n a_j r_{ij} \leq R_i \quad i = 1, \dots, m \quad (13)$$

$$\sum_{k=j}^n a_k r_{ik} > 0 \Rightarrow \sum_{k=j}^n a_k r_{ik} \leq R_i - \sum_{k=1}^{j-1} p_k r_{ik} \quad i = 1, \dots, m, j = 2, \dots, n \quad (14)$$

$$a_j \in \{0, \dots, b_j\} \quad j = 1, \dots, n \quad (15)$$

Denote the set of all allocations a that satisfy restrictions (13), (14) and (15) by A_b .

Restriction (14) states that if requests were allocated for products j, \dots, n the allocated capacity should not be greater than the available capacity minus

the amount protected for the higher nesting products $1, \dots, j-1$. Since nested booking limits b are given here, we can easily compute nested protection levels p_1, \dots, p_{n-1} and determine a priori for which products $l = j, \dots, n$ no request can be accepted anyway because $R_i - \sum_{k=1}^{l-1} p_k r_{ik} \leq 0$ for some i . Therefore, we reduce the size of the given problem by setting

$$n \leftarrow \max \left\{ j \in \{1, \dots, n\} \left| \sum_{k=1}^{j-1} p_k r_{ik} \leq R_i \right. \right\} \quad (16)$$

Let $d \in D$ be any realization of demand. Define $u_j = \max \{d_j, b_j\}$, $j = 1, \dots, n$. If $d \in A_b$, $v_b(d)$ can easily be computed: $v_b(d) = \sum_{j=1}^n d_j v_j$. Otherwise, we can use the following integer linear program to overestimate $v_b(d)$:

$$\max \sum_{j=1}^n a_j v_j \quad (17)$$

s. t.

$$\sum_{k=j}^n a_k r_{ik} \leq R_i - \sum_{k=1}^{j-1} p_k r_{ik} \quad i = 1, \dots, m, j = 1, \dots, n \quad (18)$$

$$a_j \in \{0, \dots, u_j\} \quad j = 1, \dots, n \quad (19)$$

Define

$$v_b^+(d) = \begin{cases} \sum_{j=1}^n d_j v_j & d \in A_b \\ \max \sum_{j=1}^n a_j v_j \text{ s. t. (18), (19)} & \text{otherwise} \end{cases} \quad (20)$$

Obviously, $v_b^+(d) \geq v_b(d)$. Note that $v_b^+(d) \leq v_b^+(b)$ for all $d \in D_b$ since more than b_j requests will not be accepted anyway.

3.3 Underestimating $v_b(d)$

The mixed integer program (MIP) to underestimate $v_b(d)$ is a bit more complex. It is apparently not sufficient to replace \max by \min in (17), because that will result in the trivial allocation $a_j = 0$, $j = 1, \dots, n$. But the decision maker will of course allocate as much demand as possible, i. e. (s)he will not stop to accept requests from any demand sequence unless

- there is no more demand.
- the booking limit is reached.
- the capacity is exhausted.

In the following MIP, we use a binary indicator $x_j \in \{0, 1\}$, $j = 1, \dots, n$ for the first two cases, and analogously an indicator $x_{ij} \in \{0, 1\}$ for the latter:

$$\min \sum_{j=1}^n a_j v_j \quad (21)$$

s. t.

$$\sum_{k=1}^{j-1} p_k r_{ik} + \sum_{k=j}^n a_k r_{ik} \leq R_i \quad i = 1, \dots, m, j = 1, \dots, n \quad (22)$$

$$p'_j \geq a_j \quad j = 1, \dots, n-1 \quad (23)$$

$$p'_j \geq p_j \quad j = 1, \dots, n-1 \quad (24)$$

$$p'_j \leq q_j p_j + (1 - q_j) M \quad j = 1, \dots, n-1 \quad (25)$$

$$p'_j - a_j \leq q_j M \quad j = 1, \dots, n-1 \quad (26)$$

$$\sum_{k=1}^{j-1} p'_k r_{ik} + \sum_{k=j}^n a_k r_{ik} \geq x_{ij} (R_i + \varepsilon - r_{ij}) \quad i = 1, \dots, m, j = 1, \dots, n, r_{ij} > 0 \quad (27)$$

$$a_j \geq x_j u_j \quad j = 1, \dots, n \quad (28)$$

$$x_j + \sum_{\substack{i=1 \\ r_{ij} > 0}}^m x_{ij} \geq 1 \quad j = 1, \dots, n \quad (29)$$

$$a_j \in \{0, \dots, u_j\} \quad j = 1, \dots, n \quad (30)$$

$$p'_j \geq 0 \quad j = 1, \dots, n-1 \quad (31)$$

$$q_j \in \{0, 1\} \quad j = 1, \dots, n-1 \quad (32)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, j = 1, \dots, n, r_{ij} > 0 \quad (33)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n \quad (34)$$

a_1, \dots, a_n represents the allocation whose revenue is minimized by (21). p'_j, q_j, x_{ij}, x_j are additional decision variables with the domains given by (31) to (34). M is a “large” and ε is a “small” constant; precise bounds will be given in the following.

(23) to (26) define that $p'_j = \max \{a_j, p_j\}$, $j = 1, \dots, n-1$: From (23) and (24), we have $p'_j \geq \max \{a_j, p_j\}$, $j = 1, \dots, n-1$. In addition, from (25) and (26) we have:

$$q_j = 1 \Rightarrow p'_j = p_j \wedge p'_j - a_j \leq M \quad (35)$$

$$q_j = 0 \Rightarrow p'_j = a_j \wedge p'_j \leq M \quad (36)$$

where the rightmost inequalities will hold if M is “large enough”. Since

$$p'_j - a_j \leq p'_j = \max \{a_j, p_j\} \leq b_j, \quad j = 1, \dots, n-1 \quad (37)$$

we can set:

$$M = \max_{j=1, \dots, n-1} \{b_j\} \quad (38)$$

The variables x_j, x_{ij} are used to formulate linear restrictions that represent the stopping conditions described verbally above. x_j indicates if product j has exhausted its booking limit or demand (i. e. a_j has reached the upper bound d_j or b_j), x_{ij} indicates if resource i is exhausted with respect to product j . Formally:

$$x_j = 1 \Rightarrow a_j = u_j \quad (39)$$

$$x_{ij} = 1 \Rightarrow \sum_{k=1}^{j-1} \max \{a_k, p_k\} r_{ik} + \sum_{k=j}^n a_k r_{ik} > R_i - r_{ij} \quad (40)$$

The first term on the left hand side of (40) represents capacity on resource i that is either used by or protected for products $1, \dots, j-1$. This inequality is linearized by (27). $\varepsilon > 0$ should be sufficiently small to capture that “ $>$ ” was meant where “ \geq ” was written. $\varepsilon < 10^{-(k+1)}$ is a good choice, where k is the maximum number of decimal places of every $r_{ij}, i = 1, \dots, m, j = 1, \dots, n$.

Like before, define

$$v_b^-(d) = \begin{cases} \sum_{j=1}^n d_j v_j & d \in A_b \\ \min \sum_{j=1}^n a_j v_j \text{ s. t. (22) to (34)} & \text{otherwise} \end{cases} \quad (41)$$

3.4 Deriving Upper and Lower Bounds on $E[V(b)]$

Over- and underestimating $v_b(d)$ in (11) by $v_b^-(d)$ and $v_b^+(d)$, respectively, leads the following lower and upper bound LB and UB , respectively:

$$LB = \sum_{d \in D_b} P'(d) v_b^-(d) \quad (42)$$

$$UB = \sum_{d \in D_b} P'(d) v_b^+(d) \quad (43)$$

As these bounds on $E[V(b)]$ are finite sums, they can easily be computed if $|D_b|$ is small. Unfortunately, $|D_b|$ grows exponentially with n . Therefore it may be useful to choose a “small” subset $D' \subseteq D$ and compute the following bounds instead:

$$LB' = \sum_{d \in D'} P'(d) v_b^-(d) \leq LB \quad (44)$$

$$UB' = \sum_{d \in D'} P'(d) v_b^+(d) + \left(1 - \sum_{d \in D'} P'(d)\right) v_b^+(b) \geq UB \quad (45)$$

References

1. Belobaba, P. (1987): Air Travel Demand and Airline Seat Inventory Management. PhD thesis, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA
2. Bertsimas, D., de Boer, S. (2002): Simulation-Based Booking-Limits for Airline Revenue Management. Working paper, Massachusetts Institute of Technology, Cambridge, MA
3. Bertsimas, D., Popescu, I. (2003): Revenue Management in a Dynamic Network Environment. *Transportation Science* 37, 257–277
4. Chen, V., Ruppert, D., Shoemaker, C. (1999): Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming. *Operations Research* 47, 38–53
5. Cooper, W. (2002): Asymptotic Behavior of an Allocation Policy for Revenue Management. *Operations Research* 50, 720–727
6. Klein, R. (2001): Quantitative Methoden zur Erlösmaximierung in der Dienstleistungsproduktion. *Betriebswirtschaftliche Forschung und Praxis* 53, 245–259
7. McGill, J., van Ryzin, G. (1999): Revenue Management: Research Overview and Prospects. *Transportation Science* 33, 233–256
8. Simpson, R. (1989): Using Network Flow Techniques to Find Shadow Prices for Market and Seat Inventory Control. Memorandum M89-1, Dept. of Aeronautics and Astronautics, Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge, MA
9. Talluri, K., van Ryzin, G. (1998): An Analysis of Bid-Price Controls for Network Revenue Management. *Management Science* 44, 1577–1593
10. Tscheulin, D., Lindenmeier, J. (2003): Yield-Management – Ein State-of-the-Art. *Zeitschrift für Betriebswirtschaft* 73, 629–662
11. Weatherford, L., Bodily, S. (1992): A Taxonomy and Research Overview of Perishable-Asset Revenue Management: Yield Management, Overbooking and Pricing. *Operations Research* 40, 831–844

Optimisation Methods for UMTS Radio Network Planning^{*,**}

Andreas Eisenblätter¹, Armin Fügenschuh², Hans-Florian Geerdes³, Daniel Junglas², Thorsten Koch³, and Alexander Martin²

¹ Atesio GmbH, Berlin

² Technische Universität Darmstadt

³ Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

Abstract. The UMTS radio network planning problem poses the challenge of designing a cost-effective network that provides users with sufficient coverage and capacity. We describe an optimisation model for this problem that is based on comprehensive planning data of the EU project MOMENTUM. We present heuristic mathematical methods for this realistic model, including computational results.

1 Introduction

Third generation (3G) telecommunication networks based on UMTS technology are currently being deployed across Europe. Network operators face planning challenges, for which experiences from 2G GSM barely carry over. The EU-funded project MOMENTUM developed models and simulation methods for UMTS radio network design. Among others, we devised network optimisation methods that are based on a very detailed mathematical model.

MOMENTUM constitutes, of course, not the only effort to advance methods for UMTS radio network planning. In [1–3] several optimisation models are suggested and heuristics methods such as tabu search or greedy are used to solve them. Integer programming methods for planning are shown in [12], power control and capacity issues are treated in [4,11]. Many technical aspects of UMTS networks and some practice-driven optimisation and tuning rules are given in [10]. Optimisation of certain network aspects without site selection is treated in [9].

Within this article, we focus on heuristic algorithms to solve the optimisation task. Methods based directly on the mathematical mixed integer programming model presented in [5,8] will be presented in the future. The preliminary computational results obtained within MOMENTUM are very promising.

* This work is a result of the European Project MOMENTUM, IST-2000-28088

** Supported by the DFG research center “Mathematics for key technologies” (FZT 86) in Berlin.

2 Optimisation Approach

Our optimisation approach is *snapshot based*. A snapshot is a set of users that want to use the network at the same time. We consider several snapshots at once and try to find a network that performs well for these snapshots and is cost-effective at the same time. Snapshots are typically drawn according to service-specific spatial traffic load distributions.

2.1 Optimisation Model

The following decisions have to be made for planning a network:

Site Selection. From a set \mathcal{S} of potential *sites* (roughly equivalent to roof tops where antenna masts could be placed), a subset of sites to be opened has to be chosen.

Installation Selection. At each opened site various *installations* (antenna configurations) can be employed at different antenna locations. From the set \mathcal{I} of all possible installations a subset has to be selected. The number of antennas per site is limited; *three-sectorized* sites are typical.

Mobile Assignment. For each of the users, represented by the set \mathcal{M} of *mobiles* that is possibly distributed over several snapshots, we have to decide which installation serves which mobile device. This is in practice often done on a best-server basis: each mobile is served by the installation whose signal is strongest at the mobile's location.

Power Assignment. Once the users are attached to installations, a feasible combination of power values has to be found. This includes transmission powers in uplink and downlink as well as the cells' pilot powers.

This is formulated as a MIP in [5,8], with binary variables corresponding to the first three decisions and fractional power variables p .

The coverage and capacity requirements are reflected in so-called *CIR* inequalities (Carrier-to-Interference-Ratio) that have to hold for each user. These inequalities at the core of our optimisation model follow the pattern:

$$\frac{\text{Received Signal}}{\text{Interfering Signals} + \text{Noise}} \geq \text{Threshold}$$

Using the notation from Table 1, the CIR inequality for the uplink reads:

$$\frac{\gamma_{mj}^{\uparrow} p_m^{\uparrow}}{\bar{p}_j^{\uparrow} - \gamma_{mj}^{\uparrow} \alpha_m^{\uparrow} p_m^{\uparrow}} \geq \mu_m^{\uparrow} \quad (1a)$$

The CIR inequality for the downlink is somewhat more complicated, since code orthogonality has to be considered for signals from the same cell:

$$\frac{\gamma_{jm}^{\downarrow} p_{jm}^{\downarrow}}{\gamma_{jm}^{\downarrow} (1 - \omega_m) (\bar{p}_j^{\downarrow} - \alpha_m^{\downarrow} p_{jm}^{\downarrow}) + \sum_{i \neq j} \gamma_{im}^{\downarrow} \bar{p}_i^{\downarrow} + \eta_m} \geq \mu_m^{\downarrow} \quad (1b)$$

η_m	≥ 0	noise at mobile m
$\alpha_m^\uparrow, \alpha_m^\downarrow$	$\in [0, 1]$	uplink/downlink activity factor of mobile m
ω_m	$\in [0, 1]$	orthogonality factor for mobile m
$\mu_m^\uparrow, \mu_m^\downarrow$	≥ 0	uplink/downlink CIR target for mobile m
$\gamma_{mj}^\uparrow, \gamma_{jm}^\downarrow$	$\in [0, 1]$	attenuation factors between mobile m and installation j
p_m^\uparrow	$\in \mathbb{R}_+$	uplink transmit power from mobile m
p_{im}^\downarrow	$\in \mathbb{R}_+$	downlink transmit power from installation i to mobile m
\bar{p}_j^\uparrow	$\in \mathbb{R}_+$	Total received uplink power at installation j (in the snapshot)
\bar{p}_j^\downarrow	$\in \mathbb{R}_+$	Total downlink power emitted by installation j (in the snapshot)

Table 1. Notations in CIR inequalities

2.2 Planning Data

Input data for our optimisation model is derived from the planning scenarios developed within the EU project MOMENTUM. The full contents of these scenarios are described in [7], several scenarios of them are publicly available at [13]. The scenarios contain detailed data on aspects relevant to UMTS radio network planning. The data can be classified as follows:

Radio and Environment. All aspects of the “outside” world. This includes radio propagation, UMTS radio bearers, information on the terrain (such as height or clutter data), and background noise.

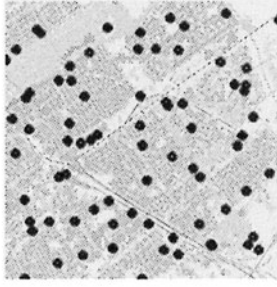
Infrastructure. All aspects that are to some extent under the control of the network operator. This includes base station hardware, antennas, potential sites and antenna locations, and radio resource management.

User Demand. All aspects related to users, such as offered services (e.g., video telephony, media streaming), user mobility, usage specifics, and traffic data.

The potential sites and installations for the planning scenario “The Hague” are shown in Fig. 1(a), the average user demand is illustrated in Fig. 1(b). Darker areas indicate higher traffic load here, the users in the snapshots are generated according to this distribution together with additional information on the used services, equipment, and mobility. The actual parameters for the optimisation model [8] and the CIR inequalities (1), in particular, are derived from the information in the planning scenarios. Table 2 gives an overview.

2.3 Preprocessing: Coverage and Capacity Analysis

Before an automatic planning process can be employed, the input data is analysed in order to detect coverage and capacity shortages. The coverage-oriented analysis is based on propagation path loss predictions for all available sites and their antenna locations. Capacity shortages are harder to detect. We use a heuristic, which is based on a tentative network design using all available sites. Employing methods similar to the ones described in [4,14], the average



(a) Potential sites and antenna configurations (installations)



(b) Traffic distribution

Fig. 1. Example of planning scenario (The Hague)

Planning Scenario	Parameter
Equipment loss, Connection loss Propagation loss, Antenna gain Usage loss (e. g. body)	Signal attenuation $\gamma_{mj}^{\uparrow}, \gamma_{jm}^{\downarrow}$
BLER requirements	
User speed	
Radio bearer	CIR targets $\mu_m^{\uparrow}, \mu_m^{\downarrow}$
User equipment, User mobility	
Radio bearer	Activity factors $\alpha_m^{\uparrow}, \alpha_m^{\downarrow}$
Clutter type	
Channel model	Orthogonality ω_m

Table 2. Derivation of parameters from the data scenarios

up- and downlink load per cell of this tentative network can be computed efficiently. If the traffic load is too high for the potential infrastructure in some regions, these can be localised as overloaded cells in the tentative network. Notice that this approach merely provides lower bounds on the achievable network up- and downlink capacity. Methods for estimating an upper bound on the network capacity are under development.

3 Heuristic Planning Methods

It turned out that solving mixed-integer program as described in its main components in Section 2.1 exactly (using for example CPLEX 8.1) takes significant time and computing resources, even for moderate sized scenarios.

Therefore, we developed various heuristic algorithms that aim at obtaining good (not necessarily optimal) solutions within reasonable running times.

The explanation of *all* these methods, including greedy-type heuristics, tabu search, simulated annealing, and evolution algorithms, is beyond the scope of this document. We restrict ourselves to the most successful one, the “Set-Covering Heuristic”. The interested reader may refer to [5,6] for the description of the other methods.

3.1 Set-Covering Heuristic

The idea of the Set-Covering Heuristic is to find for each installation $i \in \mathcal{I}$ a set M_i of mobiles that this installation can “cover” (we will explain this in more detail below). We assign a cost c_i to each of these sets M_i and then find a set $J = \{j_1, \dots, j_k\} \subseteq \{1, \dots, |\mathcal{I}|\}$ of indices such that each mobile $m \in \mathcal{M}$ is covered by at least one $M_j, j \in J$ and for which the cost $c_J = \sum_{j \in J} c_j$ is minimal. Each index in J corresponds to an installation, and we will simply select the installations that are given by J .

In order to compute the set M_i for a given installation $i \in \mathcal{I}$ we proceed as follows: First of all, we ignore all other installations $j \in \mathcal{I}, j \neq i$, that is, we assume they are not selected. We then consider each mobile $m \in \mathcal{M}$ and determine its distance $d_{m,i}$ to installation i . We define this distance to be $d_{m,i} = 1/(\gamma_{mi}^\uparrow + \gamma_{im}^\downarrow)$ if both attenuation values are non-zero (attenuation is set to zero if the corresponding pathloss exceeds a certain threshold). If the up- or downlink attenuation between mobile m and installation i is zero, this mobile can never be served by installation i . We then set $d_{m,i} = \infty$.

Let M denote the set of mobiles for which $d_{m,i} < \infty$. We initially set $M_i = \emptyset$ and sort the mobiles in M by non-decreasing values of $d_{m,i}$. According to this list we check for each mobile m , whether installation i can serve all mobiles in $M_i \cup \{m\}$ simultaneously. In the positive case we set $M_i = M_i \cup \{m\}$. The feasibility check is based on a Power Assignment Heuristic, which basically solves two systems of linear equations that arise when inequalities (1a) and (1b) are replaced with equations, see [5,6] for details.

The Power Assignment Heuristic does not only check whether installation i can serve all mobiles in $M_i \cup \{m\}$ but also finds minimal transmission powers for each mobile/installation connection in the positive case. These transmission powers are used to compute a score c_i for the resulting set M_i :

$$c_i = \sum_{m \in M_i} \lambda^\uparrow p^\uparrow + \sum_{m \in M_i} \lambda^\downarrow p^\downarrow + C_i \quad (2)$$

where the terms p^\uparrow and p^\downarrow denote up- and downlink transmission powers as returned by the Power Assignment Heuristic and C_i is the cost that is associated with installing installation i . The factors λ^\uparrow and λ^\downarrow are used to weight the transmission powers in the cost for set M_i . From iterating over the list of mobiles with $d_{m,i} < \infty$ we obtain a set M_i together with a score (or “cost”) c_i as desired; see Algorithm 1.

Algorithm 1 *Covering a set of mobiles with a given installation.*

Input: Installation $i \in \mathcal{I}$ and mobiles $M \subseteq \mathcal{M}$ that i may potentially cover.

1. Determine the mobile/installation distance $d_{m,i}$ for each mobile in \mathcal{M} .
2. Sort \mathcal{M} by non-decreasing distance to i . Denote result by M_{sorted} .
3. Set $M_{\text{return}} = \emptyset$ and $c_{\text{return}} = C_i$.
4. For each mobile $m \in M_{\text{sorted}}$ do
 - (a) Set $M' = M_{\text{return}} \cup \{m\}$.
 - (b) Use Power Assignment Heuristic to check whether installation i can serve all mobiles in M' .
 - (c) If so, set $M_{\text{return}} = M'$ and update c_{return} according to equation (2).
5. Return M_{return} and c_{return} .

Given the sets M_i and associated costs c_i for each installation, we define a set-covering problem. Let $A \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{I}|}$ denote the incidence matrix of \mathcal{M} and the M_i (i.e., $a_{ij} = 1$ if and only if mobile i is in M_j) and introduce binary variables $x_j, j = 1, \dots, |\mathcal{I}|$ that are set to one if set M_j is selected and to zero otherwise. The set-covering problem then reads as follows:

$$\min \left\{ \sum_{i \in \mathcal{I}} c_i x_i \mid Ax \geq 1, x \in \{0, 1\}^{|\mathcal{I}|} \right\} \quad (3)$$

Notice that in the above description we implicitly assume that $\bigcup_{i \in \mathcal{I}} M_i = \mathcal{M}$. If this is not the case we simply replace \mathcal{M} by $\bigcup_{i \in \mathcal{I}} M_i$.

As stated earlier, each set M_i is in direct correspondence with an installation $i \in \mathcal{I}$. Thus, given an optimal solution $x \in \{0, 1\}^{|\mathcal{I}|}$ to (3) we simply select all installations $i \in \mathcal{I}$ for which $x_i = 1$ and install them.

The Set-Covering algorithm as described above has three problems:

- Model (3) is too simplistic: it does, for example, not take into account that installations are hosted at sites. Opening such a site requires a certain amount of money (typically much more than the cost for a single antenna) and for each site there are minimum and maximum numbers of installations that can be simultaneously installed.
- Due to the fact that we *ignore* all other installation while computing the set M_i for installation i , we also ignore potential interference from these installations. The sets M_i tend to overestimate the coverage and capacity of the installations.
- The set-covering problem as defined in (3) may not have a feasible solution. This can especially happen if traffic is high and the number of installations that are available per site is limited.

All three problems can be resolved: In the first case, the additional constraints related to sites can easily be added to (3). In the second case, we shrink the sets M_i at the end of Algorithm 1 using a “shrinkage factor” f_{shrink} . Or we impose some heuristically determined interference via a “load factor” f_{load} and require that the installation may not use more than that

percentage of its maximum load during the algorithm. We distinguish two cases if (3) is infeasible. In case f_{shrink} and f_{load} equal one we declare the input infeasible (which is true up to the assumption that we have performed an optimal mobile assignment). In case at least one of these factors is less than one we modify the factors and iterate.

3.2 Results

Using the Set-Covering Heuristic we are able to compute good solutions to large-scale real world instances. We illustrate one such result for the “The Hague” scenario mentioned in Section 2. The instance contains 76 potential sites, 912 potential installations, and 10,800 mobiles partitioned into 20 snapshots (approximately 540 mobiles per snapshot). For this instance we obtained the best result using a combination of the “heuristic interference” and “heuristic shrinking” strategies by setting $f_{\text{shrink}} = 0.7$ and $f_{\text{load}} = 0.6$.

With these modifications the Set-Covering heuristic took 66 minutes on a 1 GHz INTEL PENTIUM-III processor with 2 GB RAM to find the final installation selection. Fig. 2 depicts the solution. Fig. 2(a) shows the selected installations/antennas; the load in the network is illustrated for uplink and downlink in Fig. 2(b) and Fig. 2(c) (the light areas denote a load of about 25–30%, the darker areas have less load). Our result was evaluated using advanced static network simulation methods developed within the MOMENTUM project [14]. The methods reported at most 3% missed traffic.

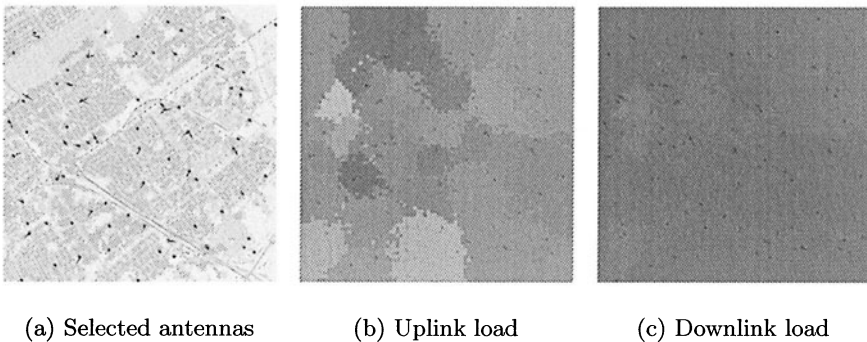


Fig. 2. Heuristic planning solution

4 Conclusion

We presented an optimisation problem of planning cost-effective UMTS radio networks. The model we use reflects many aspects of reality that are essential

for planning UMTS networks. To our knowledge, this is the most detailed and comprehensive planning model in literature. Based on this model, we have described some heuristic network planning methods that work well in practice and lead to good results.

References

1. E. Amaldi, A. Capone, F. Malucelli. Planning UMTS base station location: Optimization models with power control and algorithms. *IEEE Transactions on Wireless Communications*, 2002.
2. E. Amaldi, A. Capone, F. Malucelli, F. Signori. UMTS radio planning: Optimizing base station configuration. In *Proceedings of IEEE VTC Fall 2002*, volume 2, pp. 768–772, 2002.
3. E. Amaldi, A. Capone, F. Malucelli, F. Signori. Optimizing base station location and configuration in UMTS networks. In *Proceedings of INOC 2003*, pp. 13–18, 2003.
4. D. Catrein, L. Imhof, R. Mathar. Power control, capacity, and duality of up- and downlink in cellular CDMA systems. Tech. Rep., RWTH Aachen, 2003.
5. A. Eisenblätter, E. R. Fledderus, A. Fügenschuh, H.-F. Geerdes, B. Heideck, D. Junglas, T. Koch, T. Kürner, A. Martin. Mathematical methods for automatic optimisation of UMTS radio networks. Tech. Rep. IST-2000-28088-MOMENTUM-D43-PUB, IST-2000-28088 MOMENTUM, 2003.
6. A. Eisenblätter, H.-F. Geerdes, D. Junglas, T. Koch, T. Kürner, A. Martin. Final report on automatic planning and optimisation. Tech. Rep. IST-2000-28088-MOMENTUM-D46-PUB, IST-2000-28088 MOMENTUM, 2003.
7. A. Eisenblätter, H.-F. Geerdes, T. Koch, U. Türke. MOMENTUM public planning scenarios and their XML format. Tech. Rep. TD(03) 167, COST 273, Prague, Czech Republic, Sep. 2003.
8. A. Eisenblätter, T. Koch, A. Martin, T. Achterberg, A. Fügenschuh, A. Koster, O. Wegel, R. Wessälly. Modelling feasible network configurations for UMTS. In G. Anandalingam and S. Raghavan, editors, *Telecommunications Network Design and Management*. Kluwer, 2002.
9. A. Gerdenitsch, S. Jakl, M. Toeltsch, T. Neubauer. Intelligent algorithms for system capacity optimization of UMTS FDD networks. In *Proc. IEEE 4th International Conference on 3G Mobile Communication Technology*, pp. 222–226, London, June 2002.
10. J. Laiho, A. Wacker, T. Novosad, editors. *Radio Network Planning and Optimization for UMTS*. John Wiley & Sons Ltd., 2001.
11. K. Leibnitz. *Analytical Modeling of Power Control and its Impact on Wideband CDMA Capacity and Planning*. PhD thesis, University of Würzburg, Feb. 2003.
12. R. Mathar and M. Schmeink. Optimal base station positioning and channel assignment for 3G mobile networks by integer programming. *Ann. of Operations Research*, (107):225–236, 2001.
13. MOMENTUM Project. Models and simulations for network planning and control of UMTS. <http://momentum.zib.de>, 2001. IST-2000-28088 MOMENTUM.
14. U. Türke, R. Perera, E. Lamers, T. Winter, C. Görg. An advanced approach for QoS analysis in UMTS radio network planning. In *Proc. of the 18th International Teletraffic Congress*, pp. 91–100. VDE, 2003.

Distribution Planning Problem: A Survey

Bilge Bilgen, Irem Ozkarahan

Department of Industrial Engineering, Dokuz Eylul University, 35100,
Izmir/TURKEY

Abstract. In this paper we present a review for the distribution planning problem. We position the contribution in a framework that classifies the related research in terms of decision levels. These are strategic, tactical and operational levels. We also tabulate the reviewed research in terms of model types, model characteristics, solution procedures and the decision levels.

Keywords: distribution planning, supply chain management, survey.

1. Introduction

With the increasing significance of distribution function within supply chain management, distribution decisions become important. Distribution cost makes a significant contribution to total cost. For this reason, the design of distribution system is an important issue in many organizations. The distribution planning problem is a complex task, since it is composed of organizations linking a supplier to its various customer segments. It involves decisions related with location of warehouses, fleet management (vehicle routing considerations), and inventory related decisions.

There are many survey papers related with the SCM area. Of all the previously published survey papers in SCM area, only [1] gives special emphasis on the distribution planning problem. [1] reviews some of the significant work which has contributed to the solution of location problems of physical distribution management. [27] emphasize the importance of coordinated planning of procurement, production and distribution stages of SCM. [29] emphasize on the mixed integer programming (MIP) based strategic production and distribution models and global logistics. [7] review the production and distribution planning. [24] review the literature on integrated analysis of production and distribution system that explicitly considers transportation system.

In this paper we provide a literature survey of the distribution planning problems. We also provide a broad classification scheme. While production planning problems is of a tactical nature, distribution planning problems is more operational characteristics. The sources used for our survey study consists of scientific journals and textbooks. We exclude proceeding papers, master thesis, doctoral dissertations, and working papers. The articles are all related to the distribution planning problem. The keywords used are distribution planning, supply chain management.

The paper is organized as follows. In section 2, we review the literature in terms of relevant decision levels. In the last section we draw conclusions.

2. Review of the Related Research

We subdivide the field in terms of the relevant decision levels. The strategic level deals with the long-term decisions including DCs location. The tactical level deals with the medium term decisions including source-distribution allocation, customer allocation, plant capacity and distribution-client allocation. Operational level deals with the short term decisions, including the vehicle routing [3]. Table 1 presents a classification of the reviewed literature in a chronological order.

2.1 Strategic Models

[10] make an important contribution to the literature by presenting a new method for the solution of the problem addressed the optimal location of distribution centers (DCs). They present an algorithm based on the Benders' decomposition for solving the multi-commodity distribution system design problem. The model comprises a number of plants, DCs and customer zones whose demand are known and fixed and that must be assigned to the DCs

[25] have examined the effect of location/routing cost structures, potential number of depot sites and the spatial arrangement of customers on the performance of three alternative location routing heuristics. The research shows that the performance of alternative location routing procedures is affected by various key environmental factors.

[8] makes an important contribution to the literature by presenting different types of distribution networks with up to three transport stages namely, central warehouses, regional warehouses, transshipment points and nonlinear transport costs derived from freight tariffs. A general multi product distribution planning model is presented. The applicability of the proposed model is demonstrated by the three case studies.

[11] presents an optimization model that determines the structure of the distribution system. He develops a multi-period, facility location and material flow model. The main contribution of this model is the consideration of number of complicating constraints, such as moving units, a limited pool of transportation assets, and material flow requirements.

Table 1. Summary of the reviewed research

Reference Number	Objective Function ¹	Model Type ²	The number of commodities ³	Dynamic Characteristics ⁴	Location-Allocation-Distribution	Distribution-Routing	Location- Allocation Routing	Inventory-Routing	Solution Method ⁵	Hierarchical	Integrated	Decision Level ⁶	Industrial Application ⁷
[10]	C	D	M	S	+				BD		+	S	+
[4]	C	D	M	S		+			BD	+		O	TP
[28]	C	D	S	M			+		H	+		O	+
[25]	P	D	-	-			+		H	+		S	-
[8]	C	D	M	S	+				H			S	CS
[2]	C	D	M	M	+				B&B	+		T	CS
[5]	C	D	M	M			+		DP	+		O	CS
[18]	C	D	M	M		+			H	+		T	+
[26]	C	D	S	S			+		GA			O	TP
[13]	C	D	M	S	+				B&B		+	T	TP
[23]	C	D	S	S	+				DSS		+	T	CS
[15]	C	D	M	S	+				LH		+	T	TP
[20]	P	D	M	S	+				DSS			T	+
[9]	C	D	M	M			+		LH	+	+	O	TP
[6]	C	S	M	M	+				B&B			T	+
[22]	C	D	S	S	+				SA			T	TP
[14]	C	D	S	M				+	LH			T	TP
[17]	M	D	S	S		+			DSS			T	+
[21]	C	D	M	M				+	H	+		T	TP
[11]	C	D	M	M	+				B&B			S	TP
[30]	C	D	S	M		+			DP			O	+

1. *C* Cost *P* Profit *M* Multiple, 2. *D* Deterministic *S* Stochastic, 3. *S* Single *M* Multi

4. *S* Single period *M* Multi period 5. *BD* Benders Decompositon based heuristic, *H* Heuristic, *LH* Lagrangian-based Heuristic, *B&B* Branch and Bound Procedure, *DP* Dynamic Programming, *DSS* Decision Support System, *GA* Genetic Algorithm *SA* Simulated Annealing

6. *S* Strategic, *T* Tactical, *O* Operational 7. *CS* Case Study, *TP* Test Problems

2.2 Tactical Models

[19] develop a model for the analysis of the logistics operations at DowBrand Inc., that identifies a logistic cost savings opportunity of \$1.5 million per year. The authors develop an optimization-based Decision Support System (DSS) for designing two echelon, multi-product distribution systems. It is a deterministic formulation that considers minimization of fixed cost for establishing central and regional DCs, and variable cost for serving customers. Within the constraints, the model considers demand limits, feasible system configuration constraints.

[12] address the two-stage multi-commodity distribution planning problem. The network comprises a number of manufacturing plants, DCs and a number of customers. This problem is formulated as MIP and solved by branch and bound procedure. Later again, [13] study on the same problem by adding two complicating issues: consideration of single sourcing constraints and lot tracing.

[2] develop a MILP for minimizing logistics cost at Netherlands Car BV. The model aims to determine the ordering dates and quantities of purchase parts, given constraints on demand, transportation, inventory levels and packaging. The authors reformulate the large scale model by reducing the number of decision variables, the number of periods, the number of products so that it could be solved using branch and bound method within a reasonable time. Main concentration is given to reducing the model complexity.

[18] address the problem of optimization for the short term distribution planning in an Indian electronics firm. They present a multi-period, minimum cost network flow model for each product item. The formulation addresses the specification of the quantities to be dispatched from each production centre to the various DCs in each week of the month. They also present a real world case study to illustrate the implementation and actual use of the model. The unique feature of their model is that it can backlog the demand at any DC in any week to later in the month. A system has been developed which adapts a hierarchical planning strategy for decision making.

[20] propose a multi-disciplinary system design framework. This framework enables flexibility to analyze a wide range of supply chain network design problems. It provides a comprehensive DSS tool for distribution network planning. The main contribution of this paper is the verification the framework's large-scale capability and potential application in industry.

[23] develop an integrated methodology for planning and analysis of a petroleum distribution. The aim of the paper is to focus on reconfiguration of an existing supply chain network. A case study is presented to demonstrate the application of proposed methodology. Data envelopment analysis is used to analyze the current operations. Branch and bound procedure is used for selection of distribution facilities and allocation of resources to the facilities. This paper differs from the literature in terms of consideration of nonparametric component for performance data analysis within the developed methodology in addition to optimization based modeling component.

[15] develops a MIP based model for the distribution network design problem. The objective is to minimize the total variable cost and the total fixed costs related

with opening warehouses. The decisions to be determined include: location of facilities and specification of expected throughput levels for the facilities depending on their assigned distribution strategy. The distinguishing feature of this work is that, it demonstrates the large scale capability of the framework and verifies its potential benefits in the supply chain industry.

[14] develop a MIP model for the multi-period inventory distribution problem. The formulation addresses the problem of determining vehicle schedules and delivery quantities for the retailers. The objective is the minimization of transportation cost and inventory holding cost at retailers. A solution procedure based on a lagrangean relaxation is presented.

[22] addresses the supply network problem where resource inputs are constrained to achieve performance goals for the redesigned distribution system. The author proposes simulated annealing (SA) based heuristic in which has two phases. After the best sets of DCs are selected open, the customer assignments and resource assignments are performed. This paper makes important contributions to the literature in terms of performing the SA heuristic and the setting of the unique problem structure.

[6] present both the deterministic and stochastic models of strategic planning for logistic operations in oil industry. They implement stochastic programming solutions for a multi-period supply, transportation and distribution scheduling problem under uncertainty in the product demand. Main concentration is only on the formulation and implementation of strategic and tactical level problems. The model is solved by the XPRESS suite of LP solvers.

[17] propose a DSS for the distribution planning problem. This model differs from the literature by considering bicriteria model. Their objective involves minimization of delivery time versus maximization of volume delivered. The applicability of the proposed model is demonstrated by a case study for agricultural power fuels.

[21] presents a marine inventory/routing problem. The objective is the minimizing total cost considering both the shipping costs (fixed and variable) and the cost of violating the specified safety stock levels. The following decisions are determined: how much to ship of each product from which origin to which destination, when and by which vessel. The major contribution of this work is in the determination of the timing source and product composition of each shipment, while accounting for the product specific storage limitations at the origins and the destinations.

[16] develop a multi-echelon distribution network design problem. The network consists of a central manufacturing plant site, multiple DCs and cross-docking sites, and a set of customer zones. The models are formulated as MIP model. SA methodology is used to tackle with this problem. It incorporates cross-docking in supply chain environment.

2.3 Operational Models

[4] propose an optimization model consider a multi commodity, capacitated distribution planning model for two stages of distribution. They developed a combined distribution/routing model aiming to determine: *i.* the number and sizes of vehicles in the fleet. *ii.* the number, location and sizes of warehouses and *iii.* the product mix at each warehouse, *iv.* plant and DC assignment. The problem is solved within an algorithm based on Benders' decomposition. Computational experiments are done on randomly generated test problems.

[28] studies on a complex multi-level production and distribution network optimization problem. The problem is modeled as a MILP formulation by using fixed charge network flows supplemented by various constraints and variables. Decisions that need to be considered are: location of bottling plants and depots, the production levels at the refineries and the plants, the stock levels at the plants and the depots, the transportation volumes on each link of the distribution network, the number of trucks and drivers, the transportation shift systems and schedules, and the customer assignments. The problem is solved for various scenarios. Projected savings can be achieved ranging 10% to 25% of the total actual distribution cost. The actual problem illustrates the applicability of this procedure.

[5] develop a MILP model aiming to redesign the distribution structure for a large oil company. They decompose the problem into the components: depot location and client assignment, cyclical delivery scheduling, actual vehicle routing. These are solved sequentially. They develop a feedback mechanism to consider stage interrelations between the sub-problems. The total logistic savings are in the range of 5-6% per year.

[26] presents a mathematical model for the design of a physical distribution system. The formulation addresses the number and locations of DCs, the fleet size and the vehicle routing problem. The cost of DC, vehicle purchase cost and transportation cost of physical distribution system are included in the objective function. The constraints in the model are: vehicle capacity constraint, single sourcing of vehicles to customers, maximum number of depot locations established in the production and distribution system, route continuity constraint, assignment constraints of customers to the locations. A solution procedure based on genetic algorithm is presented. Author also presents an experimental simulation study which investigates the effects of genetic parameters on system performance.

[9] propose an optimization model for the production and distribution planning integrating components associated with capacity management, inventory allocation and vehicle routing. Their model considers single plant, multi product, multi period logistic system in which different items are manufactured and delivered with limited available resources, for both the production system and the distribution fleet. They develop a solution procedure based on lagrangian relaxation. Two solution procedures are presented: "decoupled" and "synchronized". Both procedures are tested on various sizes of the problems so that the effectiveness of the proposed solution scheme can be displayed. Substantial advantage is provided by the synchronized approach.

[30] address the problem of trying to find transportation arrangements that combines both the composition of the vehicles as well as routing of these vehicles. The problem is formulated as MILP model and dynamic programming is employed to find optimal transportation arrangement. The implementation of the new arrangement results in a decrease in total vehicle delivery cost by over 8%. The authors also present an experimental study to carry out various sensitivity analyses of the factors that critically affect the arrangements.

4. Conclusion

In this paper we review the distribution planning problem literature that deals with the strategic, tactical and operational levels. The review is also tabulated in terms of model type, model characteristics, solution procedures, and decisions levels. We draw the following conclusions:

- Not much research has been done with stochastic models.
- The vast majority of the publications found seem to include single objective function. The criterion most often used by the researchers is either minimization of costs and maximization of profits.
- Most attention has been given to strategic and tactical issues.
- New methodologies should be developed for coping with the distribution planning problem.

References

1. Aikens CH (1985) "Facility location models for distribution planning. *European Journal of Operational Research* 22:263-279
2. Ashayeri J, Westerhof AJ, Van Alst PHEL (1994) Application of mixed integer programming to a large scale logistics problem. *International Journal of Production Economics* 36:133-152
3. Ballou RH (1992) *Business Logistics Management* Prentice Hall.
4. Bookbinder JH, Kathleen ER (1988) Vehicle routing considerations in distribution system design. *European Journal of Operational Research* 37:204-213
5. Bruggen VL, Gruson R, Salomon M (1995) Reconsidering the distribution structure of gasoline products in large oil company. *European Journal of Operational Research* 81:460-473
6. Dempster M, Pedron NH, Medova EA, Scott JE, Sembos A (2000) Planning logistics operations in the oil industry. *Journal of Operational Research Society* 51:1271-1288
7. Erengüç SS, Simpson NC, Vakharia AJ (1999) Integrated production/distribution planning in supply chains: An invited overview. *European Journal of Operational Research* 115: 219-236
8. Fleischmann B (1993) Designing distribution systems with transportation economies of scale. *European Journal of Operational Research* 70:31-42

9. Fumero F, Vercellis C (1999) Synchronized development of Production, Inventory and Distribution Schedules. *Transportation Science* 33:330-340
10. Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Science* 20:822-844
11. Gue KR (2003) A dynamic distribution model for combat logistics. *Computer & Operations Research* 30:367-381.
12. Hindi KS, Basta T (1994) Computationally efficient solution of a multiproduct two-stage distribution location problem. *Journal of Operational Research Society* 45:1316-1323.
13. Hindi KS, Basta T, Pienkosz K (1998) Efficient solution of a multi-commodity two stage distribution problem with constraints on assignment of customers to distribution centers. *International Transactions of Operational Research* 5:519-527
14. Kim JU, Kim YD (2000) A Lagrangian approach to multi-period inventory/distribution planning. *Journal of the Operational Research Society* 51:364-370
15. Jayaraman V (1998) An efficient heuristic procedure for practical sized capacitated warehouse design and management. *Decision Sciences* 29:729-745
16. Jayaraman V, Ross A (2003) A simulated annealing methodology to distribution network design. *European Journal of Operational Research* 144:629-645
17. Murthy I, Olson DL, Ross A, Venkataraman M (2001) Bicriterion distribution planning for agricultural power fuels. *Infors Journal* 39:4-16
18. Prasad HVV, Sankaran JK (1996) Optimization based Distribution Planning for consumer electronic items. *Journal of Operational Research Society* 47:895-905
19. Robinson EP, Gao LL, Muggenborg SD (1993) Designing an integrated distribution system at DowBrands, Inc. *Interfaces* 23:1993
20. Robinson EP, Satterfield RK (1998) Designing distribution systems to support vendor strategies in supply chain management. *Decision Sciences* 29:685-706
21. Ronen D (2002) Marine inventory routing: shipments planning. *Journal of Operational Research Society* 53:108-114
22. Ross AD (2000) A two-phased approach to the supply network reconfiguration problem. *European Journal of Operational Research* 122:18-30
23. Ross AD, Venkataraman MA, Ernstberger KW (1998) Reconfiguring the supply network using current performance data. *Decisions Sciences* 29:707-728
24. Sarmiento M, Nagi R (1999) A review of integrated analysis of production and distribution systems. *IIE Transactions* 31:1061-1074
25. Srivastava R, Benton WC (1990) The location-routing problem: consideration in physical distribution system design. *Computer & Operations Research* 17:427-435
26. Su CT (1998) Locations and vehicle routing designs of physical distribution systems. *Production Planning and Control* 9:650-659
27. Thomas DJ, Griffin PM (1996) Coordinated supply chain management. *European Journal of Operational Research* .94:1-15
28. Van Roy T (1989) Multi-level production and distribution planning with transportation fleet optimization. *Management Science* 35:1443-1453
29. Vidal CJ, Goetschalckx M (1997) Strategic production-distribution models: a critical review with emphasis on global supply chain models. *European Journal of Operational Research* 98:1-18
30. Zhao QH, Wang SY, Xia GP, Lai KK (2003) Designing optimal routing strategies for a manufacturer: a case study. *Production Planning and Control* 14:33-41

Artikelanordnungsmuster bei Mann-zur-Ware-Kommissionierung

Karl Dörner¹, Michael Reeh¹, Christine Strauss¹ und Gerhard Wäscher²

¹ Institut für Betriebswirtschaftslehre, Universität Wien
{karl.doerner,christine.strauss}@univie.ac.at, Michael.Reeh@gmx.net

² Fakultät für Wirtschaftswissenschaft, Management Science,
Otto-von-Guericke-Universität Magdeburg
Gerhard.Waeschler@ww.uni-magdeburg.de

Zusammenfassung Der vorliegende Beitrag präsentiert die Ergebnisse einer Analyse zur Wahl eines geeigneten Artikelanordnungsmusters (Längs-, Quer-, Radialanordnung) bei gegebenen, in der Praxis häufig anzutreffenden Problemparametern. Die numerischen Ergebnisse zeigen für ein gegebenes Lagersystem, dass die Festplatzlagerung einem chaotischen Lager vorzuziehen ist und dass durch die Wahl bestimmter Artikelanordnungsmuster in Abhängigkeit von der verwendeten Routingstrategie Vorteile durch reduzierte Kommissionierwege erzielt werden können.

1 Einführung

Das Kommissionieren ist eine Lagerhausfunktion, welche die Entnahme von gelagerten Artikeln und ihr Zusammenführen gemäß vorgegebener Kundenaufträge zum Gegenstand hat. Bei einem Mann-zur-Ware-Kommissioniersystem sind die Artikel typischerweise in parallelen Gängen angeordnet, und das Kommissionierpersonal entnimmt die durch Kommissionieraufträge vorgegebenen Waren auf einem Rundweg an den jeweiligen Stellplätzen. Grundsätzlich lassen sich drei operative Entscheidungsfelder mit starken Interdependenzen identifizieren: (1) Artikelanordnung in Abhängigkeit von Artikelzugriffshäufigkeiten und Artikelvolumen (Storage), (2) Zusammenfassung mehrerer Kundenaufträge zu Kommissionieraufträgen (Batching) und (3) Bestimmung eines Wegs, den der Kommissionierer durch das Lager nimmt, um die Artikel entsprechend des Kommissionierauftrags zu entnehmen (Routing) [1,11].

Dieser Beitrag berücksichtigt sämtliche Entscheidungsfelder und analysiert die Eignung unterschiedlicher Artikelanordnungsmuster (Längs-, Quer-, Radialanordnung) bei verschieden ausgeprägten Artikelzugriffshäufigkeiten, variierter Gassenanzahl und verschiedenen Routingverfahren („Largest Gap“ und „S-Shape“). Als weitere Einflußgrößen werden das Fassungsvermögen des Kommissioniergeräts, der Auslastungsgrad des Lagers und die Größe der Kundenaufträge berücksichtigt. Ferner wird als Auftragsbildungsverfahren ein savingsbasiertes Batchingverfahren angewendet [3].

Nach einer Beschreibung der Grundstruktur von Mann-zur-Ware-Kommissioniersystemen werden in Kapitel 2 die in der Analyse verwendeten Verfahren jeweils für die drei operativen Entscheidungsfelder erläutert, der Artikelanordnung, der Auftragsbildung und der Tourenplanung. Kapitel 3 beschreibt den Aufbau der numerischen Experimente, deren Zielsetzungen und das Testdesign. Kapitel 4 beinhaltet die Ergebnisse und ihre Interpretation für die hier behandelten Routingverfahren „S-Shape“ und „Largest Gap“, sowie die Formulierung grober Entscheidungsregeln, die unter vergleichbaren Voraussetzungen als Entscheidungsunterstützung herangezogen werden können.

2 Problemstellung und Lösungsverfahren

2.1 Grundstruktur von Mann-zur-Ware-Kommissioniersystemen

Ein in der Praxis häufig anzutreffender Lagertyp besitzt typischerweise einen Blockaufbau und geht von einem rechteckig angelegten Entnahmebereich mit nur je einem Quergang an der Front- und Rückseite aus (Abb. 1). Der Zu- und Abgang vom Lager wird auch I/O-Punkt genannt und befindet sich in der unteren Ecke gegenüber dem Gang zwischen den ersten beiden Regalzeilen. Der Kommissionierer beginnt an dieser Stelle die Kommissioniertour und kehrt am Ende hierher zurück.

Der eigentliche Kommissioniervorgang basiert auf einer Reihe von Kundenaufträgen, die aus einzelnen Positionen bestehen. Eine Position bezeichnet einen Artikel(typ) sowie die vom Kunden gewünschte Menge. Es wird davon ausgegangen, dass die Menge der Kundenaufträge vorab bekannt ist (off-line Problem). Aus der Menge der Kundenaufträge werden Kommissionieraufträge (Pick-Listen) erstellt, die der Kommissionierer am Beginn einer Tour übernimmt und aus der die Menge der Artikel, deren Stellplätze und die Entnahmereihenfolge ersichtlich ist. Er benützt ein Kommissioniergerät bestimmter Größe um die entnommenen Artikel zu transportieren und mehrere Stellplätze aufsuchen zu können, bevor er sich wieder zum I/O-Punkt begibt. Es wird ferner angenommen, dass die Gänge einerseits schmal genug sind, so dass der Kommissionierer links und rechts der Gänge die Waren ohne nennenswerte Positionsänderung entnehmen kann, andererseits auch breit genug sind, um ein Überholen oder Begegnen zuzulassen.

Als Zielsetzung wird hier die Minimierung der Strecke gewählt, die zurückgelegt werden muss, um eine gegebene Menge von Kundenaufträgen abzuarbeiten. Zur Erreichung dieser Zielsetzung kann die Planung und Ausführung der beschriebenen Abläufe in drei Entscheidungsfelder (Artikelanordnung, Auftragsbildung, Tourenplanung) gegliedert werden.

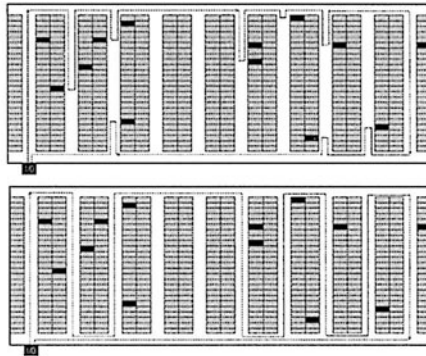


Abbildung 2. Funktionsweise von „Largest Gap“ (o.) und „S-Shape“ (u.)

ungeteilt in einem Kommissionierauftrag integriert werden sollen. Da dieser Problemtyp ebenfalls NP-schwer ist, kommt heuristischen Verfahren zentrale Bedeutung zu [2,9]. Im Rahmen dieser Analyse wird das savingsbasierte EQUAL-Verfahren angewandt [3].

Tourenplanung (Routing) Strategien, die darauf abzielen, in welcher Reihenfolge die Artikel an den einzelnen Standorten zu entnehmen sind, lassen sich in exakte und heuristische Verfahren einteilen. Exakte Verfahren bestimmen individuelle Touren, bei denen für jeden einzelnen Kommissionierauftrag die Entnahmereihenfolge bestimmt wird, während standardisierte Touren einem Routingmuster folgen. Im Folgenden werden standardisierte Touren betrachtet. Zwar ist Bestimmung einer Tour formal gesehen ein Rundreiseproblem, das in seiner Ursprungsform NP-schwer ist, jedoch ist dieses spezielle Problem wegen der Stellplatzanordnung in Gängen mit polynomialen Aufwand lösbar [8]. Es ist festzustellen, dass keine Standardverfahren zur Verfügung stehen und sich Verfahren für individuelle Implementationen für Praktiker als zu schwierig darstellen. Daher haben einfache Verfahren wie das Largest Gap-Verfahren und das S-Shape-Verfahren besondere Verbreitung gefunden [4,10].

Abbildung 2 zeigt exemplarisch ein Standardlager mit 10 Gassen mit je 50 Stellplätzen und einem Zugang links vorne. Dunkle Rechtecke repräsentieren Stellplätze, die aufgrund des Kommissionierauftrags besucht werden müssen. Das S-Shape-Verfahren (Abb. 2 unten) sieht charakteristischerweise vor, dass der Kommissionierer nicht wendet, sondern die Gasse, die er betreten hat, auf der gegenüberliegenden Seite verlässt, während das Largest Gap-Verfahren (Abb. 2 oben) Wendemanöver in fast allen Kommissioniergängen vorsieht [3].

3 Numerische Experimente – Ziel und Testdesign

Das Ziel der Analyse war die Evaluation der Anordnungsmuster, Längs-, Quer- und Radialanordnung, unter gleichzeitiger Berücksichtigung der Einflüsse aller Entscheidungsfelder (Artikelanordnung, Auftragsbildung und Tourenplanung). Konkret wurden für das Auftragsbildungsverfahren EQUAL die Auswirkungen der Routingverfahren „S-Shape“ und „Largest Gap“ auf die Wahl der Artikelanordnung analysiert. Im Gegensatz zu Petersen und Schmenner [6], die die Zusammenhänge von Anordnungsmuster, Artikelanordnung und Routingverfahren analysieren, wird hier zusätzlich die Auftragsbildung berücksichtigt. Ausgehend von einem rechteckigen Standard-Mann-zur-Ware-Kommissioniersystem mit einer Lagerkapazität von 500 Stellplätzen werden drei unterschiedliche Layouts des Lagers analysiert, die durch die Anzahl der Gassen (5, 10 und 25 Gassen mit je 50, 25 bzw. 10 Artikel pro Seite in jeder Gasse) bestimmt werden. Zur Evaluation der Längs-, Quer- und Radialanordnung wurde ein Problemgenerator implementiert, der über zwei unabhängige Zufallszahlenströme Kundenaufträge und Artikelanordnungen erzeugt. Aus der paarweisen Kombination von 10 verschiedenen Kundenauftragsbeständen und 10 verschiedenen Artikelanordnungen ergeben sich jeweils 100 Probleminstanzen [7]. Folgende Einstellungen der Problemparameter wurden getestet:

- Anzahl Stellplätze: 500
- Größe der Kundenaufträge: U [5, ..., 25]
- Gassenanzahl: 5, 10 und 25
- Artikelzugriffshäufigkeit: x^3 (60% der Zugriffe auf 20% der Artikel)
- Lagerauslastung: 80%, 90% und 100%
- Kapazität des Kommissioniergeräts: 30, 45, 60 und 75 Artikel
- Größe der Auftragsbestände: 10, 25, 50 und 100 Kundenaufträge

Für jedes der beiden Routingverfahren werden neun Ausprägungen der betrachteten Zielgröße (Länge der Kommissioniertour) angegeben, welche jeweils die Kombination einer von drei verschiedenen Lagerformen (5, 10 und 25 Gassen) mit einer von drei verschiedenen Artikelanordnungsmuster (Längs-, Quer- und Radialanordnung) bewerten. Da jede dieser neun Ausprägungen jeweils eine Aggregation über drei verschiedene Auslastungsgrade des Lagers, vier verschiedene Kapazitäten des Kommissioniergeräts und vier verschiedene Größen von Auftragsbeständen darstellt, wurden je Routingverfahren daher 43.200 Probleminstanzen bewertet; jede Ausprägung bedeutet also eine Aggregation über 4.800 Probleminstanzen. Als Messgröße dient die durchschnittliche relative Verbesserung der Ergebnisse, die gegenüber chaotischer Artikelanordnung erreicht werden.

4 Ergebnisse

Die Ergebnisse der Simulationsuntersuchungen sind jeweils getrennt nach den beiden Routingverfahren „Largest Gap“ und „S-Shape“ (Abb. 3) dargestellt.

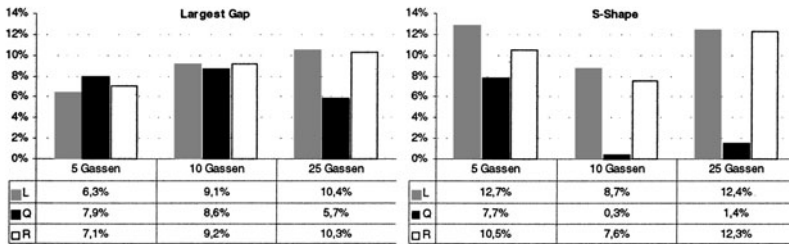


Abbildung 3. Mittlere, relative Wegeinsparung durch Artikelanordnungsmuster gegenüber chaotischem Lager bei Verwendung von „Largest Gap“ (links) bzw. „S-Shape“ (rechts) als Routingverfahren

Das relative Verbesserungspotenzial, das gegenüber chaotischer Stellplatzvergabe im Mittel erzielt werden kann, wird durch die Säulenhöhen visualisiert.

4.1 Artikelanordnung bei Routingverfahren „Largest Gap“

Bei der Anwendung des Largest Gap-Verfahrens sind bei einem Lager mit 5 Gassen fast 8% relative Einsparung bei Queranordnung gegenüber einer chaotischen Stellplatzvergabe möglich. Der Vorteil von Largest Gap, das gänzliche Traversieren einer Gasse zu vermeiden, bringt in Kombination mit langen Gassen und einer Queranordnung ein etwas höheres Einsparungspotenzial als die Längsanordnung mit 6,3% und die Radialanordnung mit 7,1%. Bei 10 Gassen zeigen alle drei Anordnungsmuster annähernd gleiches Verbesserungspotenzial von ungefähr 9% gegenüber chaotischem Lager. Bei 25 Gassen und gleichzeitig abnehmender Gassenlänge bieten querangeordnete Artikel nur noch eine Verbesserung von 5,7%, da der Kommissionierer gezwungen ist, jede Gasse vom vorderen Quergang aus zu betreten und deshalb jeweils ein Wendemanöver durchführt. Die Wegstrecke, die eventuell im Inneren der sehr kurzen Gasse eingespart werden kann, verliert im Verhältnis zum Wendemanöver an Bedeutung. Hingegen ermöglichen längs- und radialangeordnete Artikel auch hier jeweils Wegeinsparungen von etwas mehr als 10%.

4.2 Artikelanordnung bei Routingverfahren „S-Shape“

Da das S-Shape-Verfahren vorsieht, jede Gasse, in der ein Artikel zu kommissionieren ist, zur Gänze zu durchlaufen, profitiert dieses Verfahren von der Längsanordnung mit über 12% Wegeinsparung besonders bei ungerader Gassenanzahl, da die Wahrscheinlichkeit relativ hoch ist, dass in der letzten Gasse keine Artikel zu kommissionieren sind und somit das Betreten dieser letzten (ungeraden) Gasse und das damit verbundene Wendemanöver wegfällt. Bei 10 Gassen führt der Wegfall des Kommissionierens in der letzten Gasse oft zu einem Wendemanöver, das einen Teil der Wegersparnis wieder egalisiert. Die Queranordnung erweist sich in Verbindung mit „S-Shape“ verglichen mit den

anderen Anordnungsmustern am wenigsten effektiv. Im Vergleich zu chaotischer Einlagerung kommt es zwar bei geringer Gassenanzahl zu einer Einsparung von immerhin fast 7%, weil die Artikel mit großer Zugriffshäufigkeit unmittelbar beim vorderen Quergang liegen und der Kommissionierer schon nach kurzer Strecke wenden kann. Im Gegensatz dazu ist bei zufälliger Stellplatzvergabe die Wahrscheinlichkeit höher, dass Artikel zu kommissionieren sind, die wesentlich weiter im Inneren der Gasse liegen. Bei fixer Anzahl der Stellplätze bedeutet eine höhere Gassenanzahl gleichzeitig eine entsprechend verringerte Gassentiefe. Bei 25 Gassen beträgt die Gassentiefe nur noch 10 Stellplätze (statt 50 Stellplätzen bei 5 Gassen) und verringert somit das Wegeinsparungspotenzial deutlich auf 1,5%.

5 Schlussfolgerungen und Resümee

Die vorliegende Arbeit analysiert die Vorteilhaftigkeit von Artikelanordnungsmustern, nämlich der Längs-, Quer- und Radialanordnung, im Vergleich zu chaotischer Stellplatzvergabe an Hand eines exemplarischen Standard-Kommissionierlagers. Numerische Ergebnisse mit unterschiedlichen Kundenauftragsstrukturen, Auslastungsgraden, und Kapazitäten zeigen unter Berücksichtigung der Auftragsbildung das Verbesserungspotenzial gegenüber chaotischer Einlagerung getrennt nach zwei in der Praxis weit verbreiteten Routingverfahren, dem Largest Gap und dem S-Shape-Verfahren, dass Festplatzlagerung einem chaotischen Lager prinzipiell vorzuziehen ist. Ferner machen sie deutlich, dass das Ausmaß der Vorteilhaftigkeit von unterschiedlichen Anordnungsmustern bei einem Lagersystem des hier betrachteten Typs von dem angewendeten Routingverfahren abhängig ist. Bei Verwendung von „Largest Gap“ in eher schmalen Kommissionierläger mit wenigen, aber langen Gassen erweist sich die Queranordnung etwas vorteilhafter, während in breiten Kommissionierläger mit vielen, aber kurzen Gassen die Längs- oder Radialanordnung vorzuziehen ist. Ferner kann gezeigt werden, dass sich bei Verwendung von „S-Shape“ die Längsanordnung in allen untersuchten Lagerlayouts den anderen beiden Mustern gegenüber überlegen erweist, hingegen die Queranordnung den beiden anderen Anordnungsmustern deutlich unterlegen ist.

Generell kann festgestellt werden, dass Anordnungsmuster zu Wegeinsparungen zwischen 6,3% und 12,7% führen, mit Ausnahme der Queranordnung bei „S-Shape“ und geringer Gassentiefe. Die Längs- und Radialanordnung scheinen robuster als die Queranordnung zu sein, da sie meist entweder gleiche oder bessere Ergebnisse als die Queranordnung liefern.

Aus den vorliegenden Ergebnissen lassen sich als grobe Entscheidungsregel einige Aussagen formulieren: für ein Lagersystem des hier betrachteten Typs und unter Verwendung des Routingverfahrens „S-Shape“, kann durch Längs- oder Radialanordnung bei größerer Gassenanzahl durchschnittlich eine Wegeinsparung von rund 9% bis 13% erzielt werden; eine Queranordnung nach Zugriffshäufigkeit klassifizierter Artikel hat kaum Einsparungspotenzi-

al. Erfolgt die Routenwahl hingegen nach dem Largest Gap-Verfahren, so ist nur bei wenigen, langen Gassen die Queranordnung vorzuziehen und lässt bei vergleichbarer Problemstellung eine Einsparung von rund 8% erwarten. Hingegen ist bei größerer Gassenanzahl der Längs- oder der Radialanordnung der Vorzug zu geben und eine Wegersparnis ebenfalls in einer Größenordnung von 9% bis 10% zu erzielen. Da weitere Tests gezeigt haben, dass auch bei Verwendung des äußerst einfachen, in der Praxis weit verbreiteten Batchingverfahrens, First Come First Serve, im Durchschnitt Einsparungen erzielt werden können, die rund 2 Prozentpunkte unter den hier mit dem savingsbasierten EQUAL gezeigten Ergebnissen liegen. Die Aussagen bezüglich Qualität und Vorteilhaftigkeit einzelner Anordnungsmuster im gegenseitigen Vergleich treffen daher bei First Come First Serve grundsätzlich ebenso zu wie bei Verwendung von EQUAL.

Literatur

1. Caron F., Marchet, G., Perego, A. (1998): Routing Policies and COI-Based Storage Policies in Picker-to-Part Systems. *International Journal of Production Research* 36, 713-732
2. de Koster, M.B.M., van der Poort, E.S., Wolters, M. (1999): Efficient Order Batching Methods in Warehouses. *International Journal of Production Research* 37, 1479-1504
3. Dörner, K., Reeh, M., Strauß, C., Wäscher, G. (2004): Evaluation von Artikelanordnungsmustern in der Mann-zur-Ware-Kommissionierung. Working Paper, Fakultät für Wirtschaftswissenschaft, Otto-von-Guericke-Universität Magdeburg, 2004
4. Hall, R.H. (1993): Distance Approximations for Routing Manual Pickers in a Warehouse. *IIE Transactions* 24, 76-87
5. Petersen II, C.G. (1999): The Impact of Routing and Storage Policies on Warehouse Efficiency. *International Journal of Operations and Production Management* 19, 1053-1064
6. Petersen II, C.G., Schmenner, R.W. (1999): An Evaluation of Routing- and Volume-based Storage Policies in an Order Picking Operation. *Decisions Sciences* 30, 481-501
7. Rardin, R.L., Uzsoy, R. (2001): Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics* 7, 261-304
8. Ratliff, H., Rosenthal, A. (1983): Orderpicking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research* 31, 507-521
9. Rosenwein, M.B. (1994): An Application of Cluster Analysis to the Problem of Locating Items within a Warehouse. *IIE Transactions* 26, 101-113
10. Roodbergen, K.J. (2001): Layout and Routing Methods for Warehouses. Doctoral Dissertation, Erasmus Research Institute of Management (ERIM), Erasmus University, Rotterdam (ERIM Ph.D. Series Research in Management 4, TRAIL Thesis Series No. T2001/3)
11. Wäscher, G. (2003): Order Picking: A Survey of Planning Problems and Methods. In: Dyckhoff, H., Lackes, R., Reese, J. (Hrsg.): *Supply Chain Management and Reverse Logistics*, Springer, Berlin et al. 2003, 315-339

The Impact of the Exchange of Market and Stock Information on the Bullwhip Effect in Supply Chains

Bernd Faißt, Dieter Arnold, Kai Furmans

University of Karlsruhe, Germany
Institute for Conveying Technology and Logistics,
`bernd.faisst@ifl.uni-karlsruhe.de`

Abstract. We study a supply chain whose acting objects apply local control policies. The determination of the optimal order quantity is done by discret Markov decision models. The existence of the bullwhip effect is verified analytically. While analyzing the reasons of the bullwhip effect, we focus on the lack of information exchange and the uncertainty in demand. We determine the potential for improvements within the supply chain by implementing an additional information system. The shared informations are used by the discret Markov decision models to calculate the modified order quantities on every level of the supply chain.

1 Introduction

Within supply chains the participating companies usually are managing their inventory independently from upstream or downstream companies. Locally oriented inventory policies and insufficient information exchange inevitably lead to sub-optimal performance for the whole supply chain. An advantage of a cooperative contractual relationship within a supply chain is to avoid inefficient inventory control policies, which result from a lack of co-operation due to uncertain future demand rates and inaccurate information. The companies can exploit this potential only if they are willing to coordinate their control policies with the other companies in the supply chain. Starting from an isolated company we introduce a storage control model, which will be used for the subsequent steps. The calculation of the optimal order quantity is done by discrete markovian decision models. In the next section of this paper we analyze the reasons of the bullwhip effect. We focus on the lack of information exchange in an environment with stochastic market demand. This leads to over-reactions in the calculations of the demand forecast in the companies with a great distance to the market. We can show mathematically, that under the assumption of stochastic market demands the variances of the orders of a participating company is bigger than the variances of the arriving demands. Thus the existence of the bullwhip effect within a multi-stage supply chain is verified. The impact increases with the distance of the specific company from the market. We enhance the basic model of the supply chain

by an information exchange system in order to enhance the basis of information to calculate the order quantities. It can be shown that the value of the amplification of variances of the amount of orders is decreased on every stage of the supply chain. This effect is more distinct on stages of the supply chain with a big distance from the market.

2 Storage Control Model

2.1 Modelling of a Storage System as a Stochastic Dynamic Decision Problem

We analyze a storage system with a periodic registration of the inventory. The planning horizon $T = \{1, 2, \dots, \infty\}$ is assumed to be infinite and discrete. The state space is indicated by $S = \{-\infty, \dots, 1, 2, 3, \dots, \infty\}$, where $s_n \in S$ specifies the system inventory immediately before an order. The system inventory is composed by the net inventory of the storage and the inventory of the supplier. Immediately after the stock-check one has to select the amount of orders. Considering the modelling as a markov decision problem it is not reasonable to use the amount of orders as a decision variable but the system inventory after ordering. Hence there is an activity space $A_n(s_n) = \{a_n : s_n \leq a_n < \infty\} = \{s_n, s_n + 1, \dots, \infty\}$, with $a_n \in A_n(s_n)$ declares the system inventory immediately after ordering. Considering orders not to be negative, the activity space is restricted of values greater than the system inventory before ordering. For the selected amount of orders z_n in period n results $z_n = a_n - s_n$.

The demand during period n is declared by a discrete random variable D_n with $P(D_n = x) \geq 0$, $\sum_{x=0}^{\infty} P(D_n = x) = 1$, $F(\alpha) = \sum_{x=0}^{\alpha} P(D_n = x)$. Here $F(\alpha)$ indicates the distribution function of D_n . The function $p_n(s_{n+1} | s_n, a_n)$ specifies the transition probability to state $s_{n+1} \in S_{n+1}$. Because the chosen activity a_n is defined as the system inventory after ordering, the transition function results to $p(s_{n+1} | s_n, a_n) = p(s_{n+1} | a_n)$. This is the probability for s_{n+1} is independent from s_n and only depends on a_n .

At the end of the period there is a positive or negative net inventory which is used for the evaluation of costs. The stage costs are $k(s_n, a_n) = c(a_n - s_n) + l(a_n, D_n)$, $a_n \in A(s_n)$, $s_n \in S$ where $c(a_n - s_n)$ declares the linear ordering unit costs and $l(a_n, D_n)$ the storage respectively the shortfall costs in a period with demand D_n . Hence the function $l(a_n, D_n)$ also depends on the shortfall costs π , the storage costs h and the delay of delivery ν .

2.2 Determination of Optimal System Inventory

To determine the activity at the decision times we have to find a decision rule which assigns for every state $s_n \in S_n$ in every instant $n \in T$ one unique activity $a_n \in A_n(s_n)$. Such a function $f_n : S_n \rightarrow A_n$ with $f_n(s_n) \in A_n(s_n)$,

$s_n \in S_n$ is called decision rule for period n . The sequence of decision rules $d_n = (f_1, f_2, \dots, f_N) \in \Delta_N$ is called N stage strategy, where Δ_N describes the quantity of N stage strategies.

For a stationary model with infinite stages, which is formally described by $M = [S, \{A(s), s \in S\}, p, k, \beta]_{n \in [1, \dots, \infty]}$ with finite state respectively activity spaces and discounting coefficient $\beta \in [0; 1)$ applies [1]:

1. The function $\nu_\beta(s)$ is unique limited solution of the system

$$\nu_\beta(s) = \min_{a \in A_n(s)} \left\{ k(s, a) + \beta \sum_{s' \in S} p_n(s'|s; a) \nu_\beta(s') \right\}, \quad s \in S \quad (1)$$

2. There is a stationary optimal strategy $d^* = (f^*, \dots, f^*) \in \Delta$, whose decision rules f^* choose for every $s \in S$ an activity so, that the minimum of the right hand side of the system (1), the *Bellman optimization equations* [2] is realized.

For the solution of the decision problem the system (1) has a major importance. The problem of finding decision rules which are minimizing costs is that when choosing the activity at the beginning of a certain stage the future of the process has to be considered. It would be easier to find a decision rule which minimizes the costs only for a given stage. Such a strategy is called myopic [1]. In [3] the sufficient requirements for the existence of such a strategy are formulated. Hence there is a stationary strategy $d^* = (f^*, \dots, f^*)$ with

$$a = f(s) = \begin{cases} a^* & \text{for } s \leq a^* \\ s & \text{for } s > a^* \end{cases}, \quad (2)$$

where a^* declares the optimal system inventory. The system should be filled according to this at the beginning of a period.

2.3 Backorder-Case, Model with Delay of Delivery

In the backorder-case a not satisfied demand is transferred into the next period. The arising surplus will not be satisfied until the products are available at a later time instant. Considering the delay of delivery for every order there is a deterministic time of delivery of ν periods. Therefore the arrival time of an order which was placed at the time n is $n + \nu$. The function of the storage costs in period n is

$$l(a_n, \sum_{i=n}^{n+\nu} D_i) = \begin{cases} h \left(a_n - \sum_{i=n}^{n+\nu} D_i \right) \beta^\nu & \text{for } a_n \geq \sum_{i=n}^{n+\nu} D_i \\ \pi \left(\sum_{i=n}^{n+\nu} D_i - a_n \right) \beta^\nu & \text{for } a_n < \sum_{i=n}^{n+\nu} D_i \end{cases} \quad (3)$$

For any period n the expected costs are

$$E(k(a, s)) = cv(a - s) + \beta^\nu h \sum_{z=0}^a p(\sum_{i=n}^{n+\nu} D_i = z)(a - z) + \beta^\nu \pi \sum_{z=a+1}^{\infty} p(\sum_{i=n}^{n+\nu} D_i = z)(z - a) \quad (4)$$

Now we can determine the stage costs. The minimum is calculated by examination of the equation $a^* = F_{\nu+1}^{-1} \left(\frac{\pi - c(1-\beta)/\beta^\nu}{h+\pi} \right)$ [3]. Here $F_{\nu+1}$ is the $\nu + 1$ -fold convolution of the known distribution function of the demand.

2.4 Order Policies for Independent and Non Identically Distributed Demand, Backorder-Case

In the following section we develop optimal order policies for independent and non identically distributed demand. Here we assume stationary costs, i.e. $c_n = c$, $\pi_n = \pi$, $h_n = h$ for all $n \in T$.

D_n declares the time series of the demand whose elements are independent but not necessarily identically distributed. It is however assumed that the types of distributions for all F_n are identical and known. We use the normal distribution, but the resulting optimal system inventory can also be used for different distribution types.

As the demand is not identically distributed over the periods the $\nu + 1$ -fold convolution is not constant over the planning horizon and therefore the optimal system inventory could change. Now we have one out of n dependent distributions whose average value is $\mu_{gef} = d_n + \dots + d_{n+\nu}$ and the variance is $\sigma_{gef}^2 = \sigma_n^2 + \dots + \sigma_{n+\nu}^2$.

Under the assumptions that the company does not know the parameters $d_n, \dots, d_{n+\nu}$ and $\sigma_n^2, \dots, \sigma_{n+\nu}^2$ at the time instant n , it has to estimate them for the calculation of the optimal system inventory. For the determination of an ordering policy it is assumed that the turnovers stays constant. For the prediction of the totalized average values during the delay of demand we have therefore $\sum_{i=n}^{n+\nu} d_i = (\nu + 1)\tilde{d}_n$. Furthermore we assume a constant variance of the perturbation term $\tilde{\sigma}^2 = \sigma_1^2 = \dots = \sigma_N^2$ which is estimated from existing data. Under these assumptions the target system inventory is

$$a^* = \tilde{d}_n(\nu + 1) + \tilde{\sigma}\sqrt{\nu + 1}\Phi^{-1} \left(\frac{\pi - c(1-\beta)/\beta^\nu}{h + \pi} \right), \quad (5)$$

and the order policy is $z_n^* = a_n^* - a_{n-1}^* + D_{n-1} = (\tilde{d}_n - \tilde{d}_{n-1})(\nu + 1) + D_{n-1}$. Considering the assumption of constant levels the estimation of future demands can be calculated using the floating averages $\tilde{d}_n = \frac{1}{\tau} \sum_{i=n-\tau}^{n-1} D_i$ of ordinal τ . The resulting order policy is now $z_n^* = \frac{D_{n-1} - D_{n-\tau-1}}{\tau}(\nu + 1) + D_{n-1} = \frac{\nu+\tau+1}{\tau}D_{n-1} - \frac{\nu+1}{\tau}D_{n-\tau-1}$.

3 Bullwhip Effect in Consequence of Deficiency of Information

In practice the management of a storage system has the problem that it is not known which system inventory is needed to meet future demands. If there is no exchange of information within a supply chain prediction of sales can

only be created using the order data of the succeeding company. Because every company determines its inventory according to the own prediction of demand, the companies get data, which base on several prediction processes. A funded predication about the real consumption is no longer possible and there is a risk of a relevant derivation from the effective demand of customers. This amplification of the variance as a result of the distortion of informations is formally derivated in the following section using the order policy which has been described previously.

The variance of an order policy with independent not identically distributed demand can be determined using $var(a + b) = var(a) + var(b) + 2cov(a, b)$. Because of the independence of D_{n-1} and $D_{n-\tau-1}$ the covariance equals zero. Furthermore we assumed $var(D_{n-1}) = var(D_{n-\tau-1})$. This leads to the variance of the amount of orders z_n which satisfies

$$var(z_n) = \frac{(\nu + \tau + 1)^2 + (\nu + 1)^2}{\tau^2} var(D_{n-1}) > var(D_{n-1}). \quad (6)$$

The variance of the ordering policy is greater than the variance of the demand which causes the bullwhip effect within a supply chain. Moreover we can realise the dependence of the amplification from the delay of delivery and the interval used for the prediction. A great delay of delivery forces the variance to grow, while a long interval for the prediction increases the robustness against outliers. However the longer the interval is chosen the worse is the adaptability considering a change of the demand level, which leads finally to increasing costs of storage. Therefore at the choice of τ we have to find a compromise between adaptability and robustness of the prediction.

4 Implementation of an Information Exchange System

To estimate the impact of the deficit of information we enhance the supply chain by implementing a shared information system. The exchange of informations takes place at the end of every period. Here all the companies passes their current inventory as well as the occurred demands in this period. A multi-stage supply chain enhanced with an information exchange system is shown in figure 1. An important aspect of the information exchange process is the availability of the point of sales data within the whole supply chain. These data are collected on the first stage of the supply chain (stage $i = 1$).

4.1 Order Policy

For non identically and normally distributed demand and delays of delivery the optimal system inventory for a company on stage i in the supply chain equals according to equation (5)

$$a_{n,i}^* = \tilde{d}_{n,i}(\nu + 1) + \tilde{\sigma}_i \sqrt{\nu + 1} \Phi^{-1}(k_i). \quad (7)$$

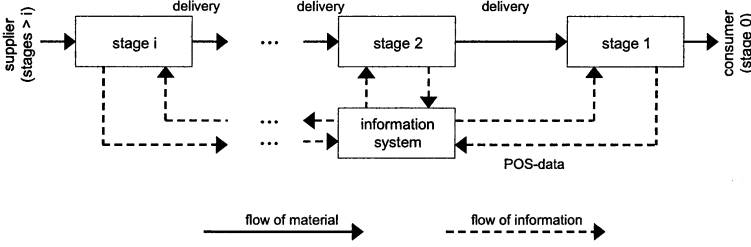


Fig. 1. Multi-stage supply chain with information exchange system

Here k_i denotes the cost structure of the company. By implementing an information exchange system for a company on stage i it is now possible to determine the optimal order $z_{n,i-1}^*$ of the company on stage $i-1$ of the supply chain. Thus the calculated order describes the known demand $D_{n,i}$. The company now only has to estimate the demands of the ν following periods applying the method of floating averages of ordinal τ . The data which can be used to perform this informations are the real market demands [4].

The optimal system inventory for a company on stage i of a supply chain applying an information exchange system can now be calculated according to

$$a_{n,i}^* = D_{n,i-1} + \tilde{d}_{n,market}\nu + \tilde{\sigma}_{market}\sqrt{\nu}\Phi^{-1}(k_i). \quad (8)$$

Applying $z_{n,i}^* = a_{n,i}^* - a_{n-1,i}^* + D_{n-1,i}$ the order policy of a company on stage i is determined by

$$z_{n,i}^* = D_{n,i-1} + \frac{\nu}{\tau} (D_{n-1,market} - D_{n-\tau-1,market}). \quad (9)$$

In the special case of immediate delivery ($\nu = 0$) and using the information exchange system according to equation (9) for the companies on stages $i = 2, 3, \dots$ it is not necessary to have a storage system to keep inventory. The orders of the ordering companies can be calculated, as long as the ordering policies are known. Thus there is no uncertainty, and a synchronised flow of material and information can be observed. Only the company on the first stage of the supply chain has to keep inventory in order to deal with the stochastic market demand.

4.2 Quantification of the Bullwhip Effect

The variance of a order policy according to equation (7) and using the assumption $var(D_{n-1,market}) = var(D_{n-\tau-1,market}) = \sigma_{market}^2$ can be calculated to

$$var(z_{n,i}^*) = var(D_{n,i-1}) + \frac{\nu^2 \sigma_{market}^2}{\tau^2}. \quad (10)$$

Using the shared informations within the whole supply chain an increase of the variance of the orders of $\frac{\nu^2 \sigma_{market}^2}{\tau^2}$ on every stage of the supply chain can be observed.

Declaring the market as the stage 0 of the market ($d_{n,Markt} = d_{n,0}$) the amplification of the order of a company on stage i of the supply chain compared to the variance of the market demand equals

$$var(z_{n,i}^*) = \sigma_{market}^2 \left(1 + i \frac{\nu^2}{\tau^2} \right). \quad (11)$$

4.3 The Impact of Information on the Bullwhip Effect

To analyse the impact of the shared information within the supply chain on the bullwhip effect it is useful to normalize the variance of the orders on a stage i of the supply chain on the variance of the market demand σ_{market}^2 . In the case of using shared informations the normalized value is calculated according to equation (11) to

$$\frac{var(z_{n,i}^*)}{\sigma_{market}^2} = \left(1 + i \frac{\nu^2}{\tau^2} \right) \equiv q_{i,info} \quad (12)$$

for $\sigma_{market}^2 > 0$. Without shared information the normalized value for the bullwhip effect equals using equation (6)

$$\frac{var(z_{n,i}^*)}{\sigma_{market}^2} = \left(\frac{(\nu + \tau + 1)^2 + (\nu + 1)^2}{\tau^2} \right)^i \equiv q_i. \quad (13)$$

Example In figure 2 the quotient

$$\frac{q_i}{q_{i,info}} = \frac{\left(\frac{(\nu + \tau + 1)^2 + (\nu + 1)^2}{\tau^2} \right)^i}{1 + i \frac{\nu^2}{\tau^2}} \quad (14)$$

is shown for $\tau = 12$. It can be observed that the delay of delivery has a big influence on the reduction of the bullwhip effect using shared informations. In general the potential of an information exchange system increases significantly for companies with a big distance to the market. The improvements for the company on the first stage of the supply chain are caused by the use of point of sales data.

5 Summary

The main content of the investigations in this paper was the analyses of a sequential supply chain, whose participants use simple mathematically developed order policies. Under the assumption of a nonexistent exchange of

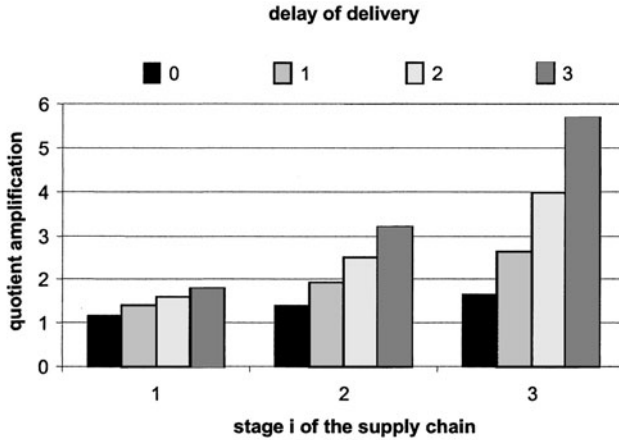


Fig. 2. Quotient $q_i/q_{i,info}$ of amplification factors, $\tau = 12$

information the bullwhip effect could be proven mathematically. It could be shown that under deficits of information the order policies lead to an amplification of variance. To weaken the bullwhip effect an information exchange system has been implemented. Through this measure the variance of ordering could be decreased significantly.

Storage control models are pretentious dynamic optimization problems. Particularly under consideration of several sequentially arranged companies in a supply chain the models show a high complexity. In further investigations we consider the problem of the complexity of the supply chains as well as the inclusion of information systems along the complete supply chain.

References

1. Girlich, H., Köchel, P., Künle, H. (1990) Steuerung dynamischer Systeme: mehrstufige Entscheidung bei Unsicherheit, Birkhäuser, Basel
2. Bellman, R., Glicksberg, I., Gross, O. (1955) On the Optimal Inventory Equation, Management Science, **1**, 83–104
3. Heyman, D.P., Sobel, M.J. [eds.](1990) Stochastic Models, North-Holland, Amsterdam
4. Faißt, B. (2003) Dynamische Effekte in Supply Chains: Der Bullwhip-Effekt als Ursache von Beständen bei Informationsdefiziten, Wissenschaftliche Berichte des Institutes für Fördertechnik und Logistiksysteme der Universität Karlsruhe, 59

Analyzing the Bullwhip Effect of Installation-Stock and Echelon-Stock Policies with Linear Control Theory

Kai Hoberg¹, Ulrich W. Thonemann¹, and James R. Bradley²

¹ Institute of Supply Chain Management, Münster University,
Universitätsstraße 14-16, 48143 Münster, Germany

² S. C. Johnson Graduate School of Management, Cornell University,
321 Sage Hall, Ithaca, NY 14853-6201, USA

Abstract. We analyze the effect of installation-stock and echelon-stock policies on order amplification in a serial two-echelon supply chain. As orders are passed through a supply chain, order oscillations and inventory oscillations are typically amplified at each echelon. This behavior is referred to as the bullwhip effect. To analyze how different inventory policies affect order and inventory oscillations we apply linear control theory. We show that the echelon-stock policy outperforms the installation-stock policy with respect to both order and inventory oscillation

1 Introduction

The costs of carrying inventory can be immense. Inventory accounts for 30% of the current assets (Stevenson [12]) while inventory carrying costs are estimated between 20% and 25% of the inventory value (Lambert and Stock [9]). Therefore, inventory-control policies that reduce inventory have become an important stream of research in operations management. Cost-based optimal inventory policies exist for many situations (e.g. Zipkin [14]). They minimize some expected cost criteria, such as the expected sum of inventory-holding and backorder-penalty cost. However, with these approaches, the dynamic behavior of a system is difficult to analyze. Therefore, we use a different approach in this paper - linear control theory. As described by Towill et al. [13], the linear control theory approach focuses on the fundamental performance of inventory policies. With linear control theory, the supply chain is described and characterized by its input-output behavior. The supply chain is represented in the frequency domain by transfer functions which can be analyzed conveniently. In recent years, the approach has been applied widely, e.g., by Dejonckheere et al. [2], to analyze forecasting approaches; by Disney and Towill [5], to study a vendor managed inventory system, and by Hafeez et al. [8], to design a steel supply chain. For an extensive overview of applications of linear control theory to inventory and supply chain management, we refer to Ortega [11].

In this paper, we apply linear control theory to analyze the effect of inventory policies on the amplification of order and inventory oscillations, which

are major topics in supply chain management. Small oscillations of customer demand are significantly increased at each echelon of a supply chain. This behavior was first identified by Forrester [6] and is referred to as the bullwhip effect by Lee et al. [10]. High order oscillations force companies to hold high levels of inventory to buffer against shortages or to provide high production capacities to react to orders quickly. Closely related to order oscillation is inventory oscillation. Inventory oscillations force companies to provide high warehouse capacity and safety stock.

The goal of our paper is to improve the understanding of how inventory policies can be represented with linear control theory and how the installation-stock policy (IS policy) and the echelon-stock policy (ES policy) affect order amplification and inventory oscillation in supply chains. The remainder of this paper is organized as follows: In Section 2, we introduce the linear-control-theoretic representation of supply chains and present our supply-chain model. In Section 3, we define performance measures and analyze the inventory policies. In Section 4, we summarize our findings.

2 Representing Inventory Policies with Linear Control Theory

In this section, we show how linear control theory can be used to model supply chains, discuss the inventory policies we analyze, and describe the assumptions of the model. For details on linear control theory, we refer to Dorf and Bishop [4] or Franklin et al. [7]. In linear control theory, the properties of a system are characterized by the transfer function. A transfer function $G(z) = \frac{Y(z)}{X(z)}$ represents the system output $Y(z)$ in relation to its input $X(z)$ in the frequency domain. The transfer function $G(z)$, when multiplied by an input $X(z)$ yields the response of the system in the frequency domain. $G(z)$ can be used to analyze how the output reacts to an input of a certain frequency and amplitude. In addition, transfer functions can be analyzed to obtain a general understanding of the system's behavior with respect to stability, dampening, steady-state accuracy, and responsiveness. The Laplace transform and z -transform are the mathematical tools of linear control theory that represent the system behavior in the frequency domain. The Laplace transform is used for a continuous-time representation of the system; the z -transform for a discrete-time representation of the system.

Figure 1 shows the structure of the two-echelon supply chain model we consider. Customer demand originates at echelon 1. Echelon 1 delivers the goods to the customers and is supplied from echelon 2, which is replenished from an outside supplier. Orders are placed periodically. The order quantity is unrestricted. The lead times are deterministic and integer. There are no capacity constraints, inventory constraints, and information delays in the supply chain. If inventory at echelon 1 drops to zero, customer demand is backordered. If inventory at echelon 2 drops to zero, goods are borrowed

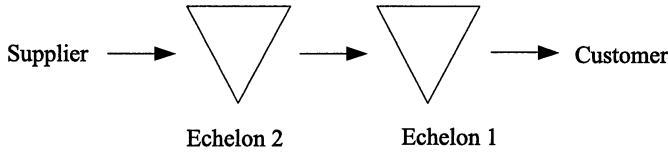


Fig. 1. Supply chain structure

from an outside source and are shipped to echelon 1 with no additional time delay. Echelon 2 restocks the outside supplier before inventory is stocked at the echelon. We note that this is a rather restrictive assumption, but the assumption is necessary to make the mathematical model tractable and is commonly used in linear control theory (e.g. Disney and Towill [5]).

Two key parameters are the lead time t_n^l and the inventory-cover time t_n^c . The lead time, i.e., the delay between the placement of an order at echelon n and the arrival of the goods at echelon n , is t_n^l . The inventory-cover time t_n^c is a factor that reflects the safety-stock requirements at echelon n . It describes the target inventory level as a multiple of the demand forecast at echelon n . The sequence of events is as follows: At the beginning of the period, the demand forecast is calculated. Then, an order is placed. Next, the shipments that were ordered t_n^l periods earlier are received. Finally, demand is received and goods are shipped to the customer.

The inventory policies we analyze are the IS policy and the ES policy. *Installation-stock policy:* The IS policy uses the sum of the on-hand inventory and pipeline inventory (i.e., the goods in transit) at echelon n as the decision state. To forecast the demand, single exponential smoothing with smoothing factor α_n ($0 < \alpha_n \leq 1$) is applied at each echelon n to the local demand at that echelon. Under an IS policy, the target installation stock TIS_n at echelon n is calculated by extending the demand forecast over the lead time and the inventory-cover time: $TIS_n = F_n \cdot (t_n^l + t_n^c)$, where F_n denotes the demand forecast at echelon n . Thus, in steady state with constant demand, the actual inventory equals the forecasted demand multiplied by the inventory-cover time. The orders O_n are calculated as the error between target installation stock and actual installation stock, $\Delta IS_n = TIS_n - IS_n$, plus the latest demand forecast F_n : $O_n = \Delta IS_n + F_n$.

Echelon-stock policy: The ES policy is an extension of the IS approach that bases order decisions on system-wide information (Clark and Scarf [1]). Demand data and inventory information of downstream echelons are shared with the upstream echelons of the supply chain. To forecast the demand at echelon n , single exponential smoothing with smoothing factor α_n is applied to the demand at echelon 1. The echelon stock is the sum of the installation stock of an installation and all downstream installations. The target echelon stock TES_n at echelon n is calculated by extending the demand forecast over the lead times and inventory-cover times of the echelon and all downstream echelons: $TES_n = F_n \cdot (\sum_{i=1}^n t_i^l + t_i^c)$. Orders are calculated as the error

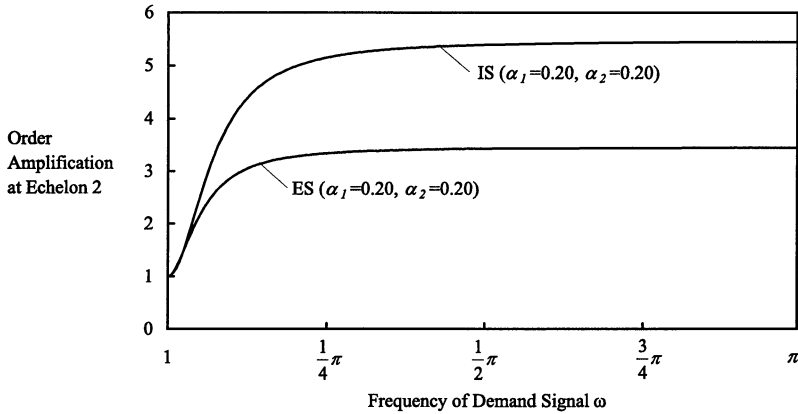


Fig. 3. Frequency-response plot of the order oscillation of IS policy and ES policy at echelon 2 with supply-chain parameters $t_1^l = 4$, $t_2^l = 4$, $t_1^c = 1$, $t_2^c = 1$

In an inventory-management context, the H_2 -norm of $G_{O_n}(z)$, $\|G_{O_n}\|_2$, is the variance of the orders at echelon n if the demand is normally distributed with variance one. The H_∞ -norm of $G_{O_n}(z)$, $\|G_{O_n}\|_\infty$, describes the worst-case demand amplification if the demand is a stationary oscillation with amplitude one. Both norms can be used as bullwhip measures of order amplification. Similarly, the H_2 -norm of $G_{I_n}(z)$, $\|G_{I_n}\|_2$, is the variance of the inventory oscillation at echelon n if demand is normally distributed with variance one. The H_∞ -norm of $G_{I_n}(z)$, $\|G_{I_n}\|_\infty$, describes the worst-case inventory oscillation at echelon n if the demand is a stationary oscillation with amplitude one.

At echelon 1, the IS policy and the ES policy are identical and there is no effect of the parameters at echelon 2 on the performance of echelon 1. Therefore, we disregard echelon 1 and focus the analysis on echelon 2. We used a test bed with 500 supply chains with different lead times t_n^l , inventory-cover times t_n^c , and smoothing factors α_n to compare the performances of the IS and ES policies. The results show that the ES policy performs significantly better than the IS policy. Table 1 summarizes the average relative improvement \bar{I} of an ES policy over an IS policy for the different performance measures. The amplification of order and inventory oscillations under normally distributed demand, i.e., $\|G_{O_2}\|_2$ and $\|G_{I_2}\|_2$, are on average 38.93% and 40.34% lower under the ES policy than under the IS policy. The maximum amplification of order and inventory oscillations, i.e., $\|G_{O_2}\|_\infty$ and $\|G_{I_2}\|_\infty$, are on average 27.34% and 31.17% lower under the ES policy than under the IS policy. In addition, the ES policy outperformed the IS policy not only on average, but for all performance measures in each of 500 supply chains analyzed.

The response of the policies with respect to any stationary demand can also be analyzed with the transfer functions. Figure 3 shows the order amplification of the policies at echelon 2 as a function of the demand frequency

Table 1. Average relative improvement of an ES policy over an IS policy

	$\ G_{O_2}\ _2$	$\ G_{O_2}\ _\infty$	$\ G_{I_2}\ _2$	$\ G_{I_2}\ _\infty$
\bar{H}	38.93%	27.34%	40.34%	31.17%

Table 2. Performance measures for IS policy and ES policy ($\alpha_1=\alpha_2=0.20$) for supply chain parameters $t_1^l = 4, t_2^l = 4, t_1^c = 1, t_2^c = 1$

	$\ G_{O_2}\ _2$	$\ G_{O_2}\ _\infty$	$\ G_{I_2}\ _2$	$\ G_{I_2}\ _\infty$
IS policy	5.09	5.44	6.35	13.08
ES policy	3.28	3.44	4.75	9.15

ω for a supply chain with parameters $t_1^l = 4, t_2^l = 4, t_1^c = 1, t_2^c = 1$. For constant demand ($\omega = 0$), the order amplification is one, i.e., there is no order amplification. As the frequency of the demand signal increases, the order amplifications of both policies increase. However, the order amplification of the ES policy is significantly lower than the order amplification of the IS policy. The frequency-response plot can be linked to $\|G_{O_2}\|_2$ and $\|G_{O_2}\|_\infty$. Since normally-distributed demand is equivalent to a signal with equal amplitudes at all frequencies, $\|G_{O_2}\|_2$ can be computed as the integral of the functions in Figure 3. In this particular example, $\|G_{O_2}\|_2$ is 5.09 for the IS policy and 3.28 for the ES policy. The worst-case response to a demand with a constant frequency, $\|G_{O_2}\|_\infty$, corresponds to the maximum amplitude of the functions in Figure 3. In this particular example, $\|G_{O_2}\|_\infty$ is 5.44 for the IS and 3.44 for the ES policy. From the frequency-response plot in Figure 4 we can see that similar results hold for inventory oscillations. However, we find that, for a few frequencies, an IS policy can be favorable to an ES policy with respect to inventory oscillation. Table 2 summarizes the performance measures of this example.

The results of our numerical experiments indicate that the ES policy clearly dominates the IS policy both in terms of order and inventory oscillations. Since inventory oscillations are lower under an ES policy than under an IS policy, the ES policy requires less safety stock than the IS policy, which in turn results in lower inventory-holding costs. Since order oscillation is also lower under the ES policy than under the IS policy, the ES policy also results in lower production costs than the IS policy.

4 Conclusion

In this paper we have used linear control theory to model IS and ES inventory policies in a two-echelon supply chain. The discrete transfer function approach we chose describes the input-output behavior of a supply chain in single equations. We have shown how different performance measures based

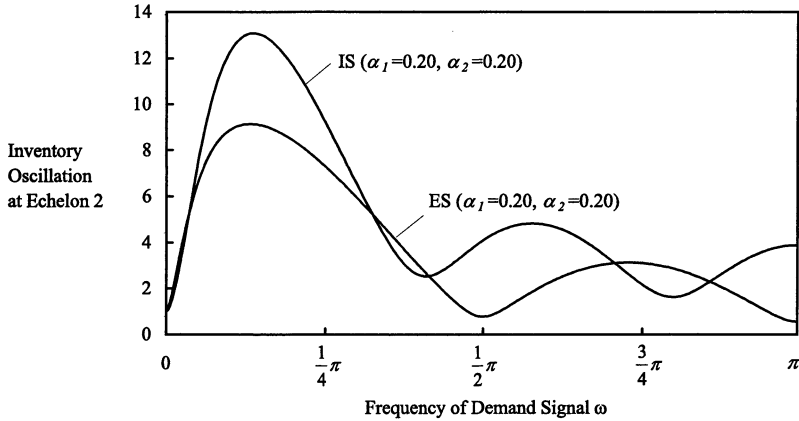


Fig. 4. Frequency-response plot of the inventory oscillation of IS policy and ES policy at echelon 2 with supply-chain parameters $t_1^l = 4$, $t_2^l = 4$, $t_1^c = 1$, $t_2^c = 1$

on Hardy-space norms can be applied to the transfer functions. Since the mathematical complexity of these performance measures is very high, general analytical results could not be obtained, but we were able to gain insights into the performance of the IS and ES policies by numerical analysis of the transfer functions. With respect to order amplification, the ES policy is superior to the IS policy. This could be observed for all reasonable smoothing factors in response to normally distributed demand, to worst-case stationary demand, and over all demand frequencies. For inventory oscillation, similar results hold with respect to normally distributed demand and to worst-case stationary demand.

In order to decrease the order amplification of an IS policy, Dejonckheere et al. [3] introduce a partial correction of the error between target and actual installation stock. However, by correcting the error over several periods, the policy's responsiveness to non-stationary demand, e.g., a step change in demand, can decrease. We leave the analysis of this dynamic behavior of supply chains in response to non-stationary demand as well as the influence of information delays in ES policies for further research.

References

1. Clark, A.J., Scarf, H. 1960. Optimal Policies for a Multi-Echelon Inventory Problem. *Management Science* 6(4), 475-490.
2. Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R. 2002. Transfer Function Analysis of Forecasting Induced Bullwhip in Supply Chains. *International Journal of Production Economics* 78, 133-144.
3. Dejonckheere, J., Disney, S.M., Lambrecht, M.R. and Towill, D.R. 2003. Measuring and Avoiding the Bullwhip Effect: A Control Theoretic Approach. *European Journal of Operational Research* 147, 567-590.

4. Dorf, R.; Bishop, R.H. 2001. Modern Control Systems, 9th ed., Upper Saddle River, NJ, Prentice-Hall.
5. Disney, S.M., Towill, D.R. 2002. A Discrete Transfer Function Model to Determine the Dynamic Stability of a Vendor Managed Inventory Supply Chain. *International Journal of Production Research* 40, 179-204
6. Forrester, J.W. 1962. Industrial Dynamics, Cambridge, Mass.
7. Franklin, G.F., Powell, J.D., Workman, M.K. 1997. Digital Control of Dynamic Systems. 3rd ed., Upper Saddle River, NJ, Prentice-Hall
8. Hafeez, M., Griffiths, M., Griffiths, J., Naim, M.M. 1996. Systems design of a two-echelon steel industry supply chain. *International Journal of Production Economics* 45, 121-130
9. Lambert, D.M. and Stock, J.R. 1993. Strategic Logistics Management. Homewood, Illinois, Irwin.
10. Lee, H.L., Padmanabhan, V., Whang, S. 1997. Information Distortion in a Supply Chain: The Bullwhip Effect. *Management Science* 43(4), 1997, 546-558
11. Ortega, M.J. 2001. Analysis of production-inventory systems by dual-models of control theoretic and discrete-event simulations. PhD Thesis, State University of New York at Buffalo
12. Stevenson, W.J. 1999. Production and Operations Management, Irwin McGraw-Hill
13. Towill, D.R., Lambrecht, M., Dejonckheere, J. and Disney, S.M. 2003. Explicit Filters and Supply Chain Design. Working Paper, To be published in: *Journal of Purchasing & Supply Management*
14. Zipkin, P.H. 2000. Foundations of Inventory Management. McGraw-Hill, Boston

Policy Approximation for the Production Inventory Problem with Stochastic Demand, Stochastic Yield and Production Leadtime

Karl Inderfurth, Christian Gotzel

Otto von Guericke University of Magdeburg, Faculty of Economics and Management, P.O. Box 4120, 39106 Magdeburg, Germany

Abstract. We consider a single-stage production inventory problem under periodic review with stochastic demand and stochastically proportional yield. Holding, shortage and ordering costs are assumed to be strictly proportional. The optimal ordering policy is known to be of a non-linear type. Linear approximation of this policy is a commonly used heuristic approach. Our focus is on the application of linear approximations under procurement leadtime conditions for an infinite horizon setting.

1 Introduction

The problem of stochastic yield is experienced in several industrial production processes. Particularly, yield uncertainty is inherent in processes that are not completely controllable such as chemical processes or semiconductor fabrication.

We consider a single-stage production problem with stochastic yield and demand. Production yield is a random fraction of the production order. This yield model is usually referred to as stochastically proportional yield. Yield and demand distributions are assumed to be stationary and independent. Procurement, holding and shortage costs are strictly proportional. Any unsatisfied demands are backlogged. We also allow for a deterministic production leadtime.

Production inventory problems with yield uncertainty as described above have been subject of a number of publications. For problems with zero leadtime it has been shown in [3, 4] that the optimal production policy is an extension of the well-known order-up-to policy for the deterministic yield case which, however, results in a non-linear procurement rule that hardly can be implemented in practice. In industry facing yield problems usually linear procurement policies are applied as we find in MRP approaches where net requirements are inflated by constant yield factors. For zero leadtime linear policy approximations have been considered by several authors (see [1, 2, 4, 6]). In extending this line of research our paper focuses on the development of most effective linear approximations in the case of production leadtime.

In the next section of our paper we formulate the optimisation problem and discuss the application of linear policy approximations. In a subsequent section, the linear approximation is extended to the condition of a non-zero procurement lead-time, and numerical results for optimising a linear procurement policy are reported. We end with some final conclusions.

2 Optimisation Model and Heuristic Approaches

In the sequel we will consider a single-stage single-item production facility with stochastically proportional yield, i.e. the output acceptable is a random fraction of the order quantity. Holding, shortage and procurement costs are assumed to be strictly proportional, there are no fixed procurement costs. A procurement lead-time is not (yet) considered. The following notation is used:

- h : holding cost per unit
- v : shortage cost per unit
- c : procurement cost per unit
- r : stochastic demand with pdf $f(r)$, cdf $F(r)$, mean μ_r and variance σ_r^2
- z : stochastic yield fraction with pdf $g(z)$, and cdf $G(z)$, mean μ_z and variance σ_z^2
- C : total cost
- x : initial inventory minus backorder
- p : procurement quantity

We start with the single-period problem. For a given inventory level x and procurement quantity p , the expected period total cost is

$$\begin{aligned} L(x, p) = & c \cdot p + h \cdot \int_0^\infty g(z) \int_0^{x+z \cdot p} (x + z \cdot p - r) \cdot f(r) \cdot dr \cdot dz \\ & + v \cdot \int_0^\infty g(z) \int_{x+z \cdot p}^\infty (r - x - z \cdot p) \cdot f(r) \cdot dr \cdot dz . \end{aligned} \quad (1)$$

In the case of multiple periods the following recursive relationship holds for the minimal cost C_n of an n -period problem.

$$C_n(x) = \min_{p \geq 0} \left\{ L(x, p) + E_{z,r} [C_{n-1}(x + z \cdot p - r)] \right\} \quad (2)$$

In [3, 4] it is shown that the optimal policy $p^*(x)$ for this problem is an extension of the order-up-to policy in case of certain yield. It can be described by

$$p^*(x) = \begin{cases} P(x) & \text{if } x \leq S^+ \\ 0 & \text{if } x > S^+ \end{cases} \quad (3)$$

where $P(x)$ is a procurement function decreasing in x with $dP/dx \leq -1$ and $P(S^+) = 0$.

The policy in Eq. (3) holds for the single-period as well as for the multi-period (and even for the infinite-period) case. For the single-period problem, the procurement level S^+ is given by a newsvendor-type solution

$$S^+ = F^{-1}(\alpha) \quad \text{with} \quad \alpha = \frac{v - c/\mu_z}{v + h}. \quad (4)$$

Obviously, S^+ is only affected by the mean yield rate, but not by yield variability. If procurement costs are only charged for procurement yield (i.e. $z \cdot p$), S^+ is completely identical to the standard newsvendor formula (see [3]). In the multi-period case, procurement level S^+ is larger than the newsvendor level and is affected by the complete yield distribution function. For this reason a myopic policy will not be optimal in a multi-period decision framework. However, when applying the single-period solution as myopic rule in the stationary infinite-horizon case, an easy approximation for S^+ - according the results for the certain yield problem - is given by omitting the procurement cost term in Eq. (4):

$$S^+ = F^{-1}\left(\frac{v}{v + h}\right) \quad (5)$$

Unfortunately, the procurement function in Eq. (3) cannot in general be given in a closed form, except for the single-period case with specific demand and yield distributions (like uniform or exponential pdf, see [3]). For uniformly distributed demand and yield rate with $r \in [0, r^+]$ and $z \in [0, z^+]$ the result in [3] requires a correction as shown in [5]:

$$P(x) = \begin{cases} \frac{1}{z^+} \cdot \sqrt{\frac{(r^+ - x)^3}{3 \cdot (1 - \alpha) \cdot r^+}} & \text{if } 0 \leq x \leq (3\alpha - 2) \cdot r^+ \\ \frac{3}{2 \cdot z^+} \cdot (\alpha \cdot r^+ - x) & \text{if } (3\alpha - 2) \cdot r^+ \leq x \leq \alpha \cdot r^+ = S^+ \end{cases} \quad (6)$$

with S^+ from Eq. (4). So, it turns out that also for very special cases $P(x)$ usually can be expected to be of a complex non-linear form. Even if the optimal procurement function could be described analytically in more general cases, such a procurement policy would not be attractive for practical applications because of its non-linearity. For that reason we will consider approximations of $P(x)$ of which the most tractable seems to be a linear one, being described by two parameters P_0 and β :

$$\hat{P}(x) = P_0 - \beta \cdot x \quad (7)$$

Hereby, we get for the respective procurement level: $\hat{S}^+ = P_0 / \beta$. In literature, different suggestions for this kind of approximation have been made in [2, 4, 6], where always \hat{S}^+ is fixed at the newsvendor level in Eq. (5) and different β -values are chosen like: (I) $\beta = 1/\mu_z$ to represent the optimal policy in the certain yield case, or (II) $\beta = \mu_z / (\mu_z^2 + \sigma_z^2)$ which has been derived by the authors in quite different ways. Unlike in these approaches, in [1] the authors fix β to $1/\mu_z$ and choose a P_0 value such that \hat{S}^+ is different from the newsvendor solution resulting in a linear function $\hat{P}_{III}(x)$ differing from the functions $\hat{P}_I(x)$ and $\hat{P}_{II}(x)$ for the above approximations. Fig.1 gives a graphical representation of how the exact order policy $P(x)$ is approximated by the different linear heuristics.

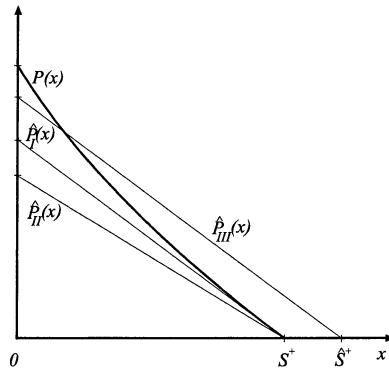


Fig. 1. Linear approximation to the non-linear optimal policy

For developing the $\hat{P}(x)$ function the problem can be reformulated using the definition of a random variable aggregating the variability of demand and yield $\eta = r - (z - \mu_z) \cdot p$ with pdf f_η (see [1]). Note that the distribution of this modified demand variable depends on the production quantity p . The expected inventory available for demand fulfilment is $y = x + \mu_z \cdot p$, so that $y - \eta = x + z \cdot p - r$ represents the end of period inventory level. Similar to the newsvendor model, the expected single-period cost function (omitting procurement cost) is

$$L(x, y) = h \cdot \int_0^y (y - \eta) \cdot f_\eta(\eta) \cdot d\eta + v \cdot \int_y^\infty (\eta - y) \cdot f_\eta(\eta) \cdot d\eta \quad (8)$$

From the first order derivative with respect to y we receive as an optimality condition

$$F_{\eta}(y) + \int_0^y (z - y) \cdot \frac{\partial f(\eta, y)}{\partial y} \cdot d\eta = \frac{v}{v + h} \quad (9)$$

which obviously deviates from the simple newsvendor solution. The integral term in Eq. (9) is due to the fact that η depends on y . However, in [1] this difference is neglected and the expression without the integral term is used to derive a myopic heuristic.

Thus, procurement function $P(x)$ is implicitly given by

$$\text{Prob}\{r - (z - \mu_z) \cdot p \leq x + \mu_z \cdot p\} = \frac{v}{v + h} \cdot \quad (10)$$

$P(x)$ now is approximated by the linear function

$$\hat{P}(x) = P_0 - \frac{1}{\mu_z} \cdot x \quad (11)$$

where P_0 is determined in such way that equation (10) holds when inserting $p = \hat{P}(x)$ and fixing x to a specific value (see [1]). Since this heuristic has proven to perform quite well, it will be chosen as basis for an extension to a setting with production leadtimes.

3 Linear Heuristics in Case of Production Leadtimes

In this section, we concentrate on the aspect of integrating a production leadtime in a linear heuristic. With a leadtime of λ , a procurement order p_t , placed in period t , will be received after λ time-periods as a quantity $z_t \cdot p_t$ of acceptable items.

Since a decision in t will replenish the inventory in $t + \lambda$, we introduce an inventory position x'_t at the beginning of period t , aggregating the present stock on hand x_t plus the yield expectation of all outstanding orders,

$$x'_t = x_t + \mu_z \cdot \sum_{i=1}^{\lambda} p_{t-i} \quad (12)$$

The respective linear procurement policy extending (11) can be written as

$$\hat{P}_t(x'_t) = \frac{S_t - x'_t}{\mu_z} \quad (13)$$

It orders up to the critical inventory level $S_t = SST_t + (\lambda + 1) \cdot \mu_r$, where SST_t represents the safety stock. Note that in the procurement leadtime case, despite the stationarity assumption, the safety stock can vary from period to period due to

yield uncertainty affected by the specific outstanding orders. Extending the zero leadtime case we get for the modified random variable η_t

$$\eta_t = \sum_{i=0}^{\lambda} r_{t+i} - \sum_{i=0}^{\lambda} (z_{t-i} - \mu_z) \cdot p_{t-i} \quad (14)$$

For the expected inventory position after the current order is received we define

$$y_t = x_t + \mu_z \cdot \sum_{i=0}^{\lambda} p_{t-i} = x'_t + \mu_z \cdot p_t \quad (15)$$

After inserting both variables into the optimality condition $\text{Prob}\{\eta_t \leq y_t\} = \nu/(\nu + h)$ we get the conditional equation

$$\text{Prob}\left\{\zeta_t := \sum_{i=0}^{\lambda} r_{t+i} - \sum_{i=1}^{\lambda} z_{t-i} \cdot p_{t-i} - z_t \cdot p_t(x') \leq x' - \mu_z \cdot \sum_{i=1}^{\lambda} p_{t-i}\right\} = \frac{\nu}{\nu + h} \quad (16)$$

which defines the non-linear procurement function. In order to get a linear approximation we have to insert

$$P_t(x') = \frac{1}{\mu_z} \cdot (SST_t + (\lambda + 1) \cdot \mu_r - x') \quad (17)$$

from Eq. (13) for $p_t(x')$ in Eq. (16) and set x' to a specific value, denoted by \hat{x} . Additionally assuming that ζ_t is (approximately) normally distributed, we can solve Eq. (16) for SST_t . Choosing a specific level \hat{x} corresponds to determining the point $(\hat{x}, p_t(\hat{x}))$ of the non-linear procurement function which will be used as intersection point for the linear policy approximation as depicted for function $\hat{P}_{III}(x)$ in Fig.1. Following this line of consideration and choosing $\hat{x} = SST_t$ (as proposed in [1] for the zero leadtime case) yields:

$$SST_t = \Phi^{-1}\left(\frac{\nu}{\nu + h}\right) \cdot \sqrt{(\lambda + 1) \cdot \sigma_r^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda} p_{t-i}^2 + (\lambda + 1)^2 \cdot \sigma_z^2 \cdot \frac{\mu_r^2}{\mu_z^2}} \quad (18)$$

where Φ stands for the standard normal cdf.

Alternatively, choosing $\hat{x} = E[x'] = SST_t + \lambda \cdot \mu_r$ results in

$$SST_t = \Phi^{-1}\left(\frac{\nu}{\nu + h}\right) \cdot \sqrt{(\lambda + 1) \cdot \sigma_r^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda} p_{t-i}^2 + \sigma_z^2 \cdot \frac{\mu_r^2}{\mu_z^2}} \quad (19)$$

which differs from Eq. (18) by the multiplier $(\lambda + 1)^2$ of the risk term $\sigma_z^2 \cdot \mu_r^2 / \mu_z^2$. In order to investigate the impact of using different approximation points \hat{x} , we parametrise this multiplier by the term γ :

$$SST_i = \Phi^{-1} \left(\frac{\nu}{\nu + h} \right) \cdot \sqrt{(\lambda + 1) \cdot \sigma_r^2 + \sigma_z^2 \cdot \sum_{i=1}^{\lambda} p_{t-i}^2 + \gamma \cdot \sigma_z^2 \cdot \frac{\mu_r^2}{\mu_z^2}} \quad (20)$$

In order to get an idea of how to choose γ best, we tested this approximation numerically over an infinite horizon under different cost, demand and yield scenarios. It turns out that the optimal level of this parameter is $\gamma=1$ which supports the choice of $\hat{x} = E[x']$ for the approximation point.

Finally, we will present some typical numerical results that show how the above approximation with the safety stock setting from Eq. (19) performs in comparison to other linear approximations. As another linear approach that we include in our comparison, we set $\beta=1/\mu_z$ and a fixed procurement level S^+ as in the no-defects (newsvendor) model as proposed in [6].

In our scenario, demand is gamma-distributed with $\mu_r=100$ and $\sigma_r=20$, yield is beta-distributed with $\mu_z=0.5$ and $\sigma_z=0.125$, holding and shortage costs are $h=3$ and $\nu=27$. A set of 3 heuristics is compared. Heuristic I with $\hat{x} = E[x']$ is represented by Eq. (19) whereas heuristic II is based on the approximation point $\hat{x} = SST$ as given by Eq. (18). For completing the comparison also the linear heuristic with a leadtime adjusted newsvendor S^+ and $\beta=1/\mu_z$, referred to here as heuristic III, is reported.

Table 1. Cost performance of different linear heuristics

λ	C_I	C_{II}	C_{III}
0	100	100	114
2	100	120	116
4	100	142	116
6	100	161	116
8	100	179	116
10	100	195	116

Numerical results are given in Tab.1. Total costs computed for heuristic I were standardised to 100 and results obtained for the other heuristics are expressed relatively to C_I . From the data it can be seen that depending on the leadtime, the choice of γ significantly impacts the cost performance of the linear approximation. It becomes apparent that heuristic II performs worst with extremely large deviations from C_I , which is a result of a safety stock level fixed too high in Eq. (18). Disregarding yield uncertainty, the leadtime adjusted newsvendor heuristic generates an insufficient safety stock level and therefore causes an increased occurrence of stockouts. Applying this heuristic yields a quite large cost deviation from C_I , however, the gap is negligibly affected by the leadtime.

4 Conclusions

Our results show that applying a linear approximation to the optimal order policy requires determining the approximation point \hat{x} which can be shown to have a considerable impact on the performance of this approach. Thus, fixing this point in a proper way is an important issue when developing a linear heuristic as shown in last section. Numerical results suggest that the approximation point should be chosen to represent the expectation of the inventory position x' . Further on, heuristics that do not account for the yield risk inherent in the individual outstanding orders provide a rather poor performance.

References

1. Bollapragada S, Morton T (1999) Myopic heuristics for the random yield problem. *Operations Research* 47(5):713-722
2. Ehrhardt R, Taube L (1987) An inventory model with random replenishment quantities. *International Journal of Production Research* 25:1795-1803
3. Gerchak Y, Vickson R, Parlar M (1988) Periodic review production models with variable yield and uncertain demand. *IIE Transactions* 20:144-150
4. Henig M, Gerchak Y (1990) The structure of periodic review policies in the presence of random yield. *Operations Research* 38(4):634-643
5. Inderfurth K (2004) Analytical solution for the single-period inventory problem with uniformly distributed yield and demand. *Central European Journal of Operations Research*, forthcoming
6. Zipkin PH (2000) *Foundations of Inventory Management*, McGraw-Hill, Boston

A Dynamic Model for Choosing the Optimal Technology in the Context of Reverse Logistics

Rainer Kleber

Otto-von-Guericke University Magdeburg, Faculty of Economics and
Management, P.O. Box 4120, 39016 Magdeburg, Germany
E-mail: rainer.kleber@ww.uni-magdeburg.de

Abstract. This paper is intended to give insights into the simultaneous technology choice and investment time problem within a dynamic deterministic framework consisting of a product life cycle and an availability cycle of returns. A Net Present Value approach is employed for solving the problem.

1 Introduction

Developing new products and setting up production facilities requires firms to choose between different technologies in order to manufacture the product. Besides quality and service aspects, this decision has an impact on direct production costs and necessary capital expenditures in building new facilities or modifying existing ones. In the context of reverse logistics, the product design decision also affects the possible options on how to deal with returned used products, and it has to be extended by the question of whether to design and produce a product for a single life time only, or in such a way that it or its components can be made capable for a second use after some recovery process. This can yield additional profit, as some of the added value will not be lost through material-recycling or disposing of the returned item. On the other side, there can be higher expenses for setting up production facilities, as well as higher production unit costs, that are caused by the necessity to add properties to the product for making it recoverable.

Although there exists an enormous amount of literature on operative issues in reverse logistics (for a literature survey see [1]), aspects of financial justification, being highly influential to investment decisions, have not been addressed in detail so far because they require both a dynamic consideration as well as the application of discounted cash flow techniques, which substantially complicates the investigation. In recent contributions to dynamic product recovery (see [2] for an overview), it is assumed that both, production and remanufacturing, facilities already existed, i.e. the technology choice has been made before. Since introducing a technology for remanufacturing may require additional investments, it remains to be seen if these investments will pay off or not. Moreover, it is not necessarily useful to set up a remanufacturing facility immediately at the time production starts, because this usually requires a considerable capital commitment while not yet having a

large number of returns available. Therefore, a decision has to be made when to introduce this process. Another question is how the possibility to hold returned items in a strategic inventory influences decision making.

The paper is organized as follows. In Section 2, three investment projects representing different environmental policies are introduced, and for each one of them the optimal project parameters are determined. The last section provides conclusions and possibilities for further research.

2 Three Investment Projects

This section deals with three basic investment projects. Three types of operational cash outflows are considered: investment expenditures, constant production, remanufacturing and disposal per unit payments as well as out-of-pocket inventory holding costs. Recovered products are regarded as a perfect substitute to newly produced items, and thus are sold to the same market. Revenues, as well as payments connected with the take back of used products are not taken into account, since both demand as well as returns are assumed to be given. Furthermore, since we are performing a long-term analysis, all operative processes are supposed to have sufficient capacity available. We consider a demand/return scenario complying with the following assumptions:

A.1 Demand $d(t)$ is assumed to be a deterministic continuously differentiable function of time showing the typically unimodal shape of a product life cycle with its maximum located at $t_d^{max} > 0$.

A.2 Returns $u(t)$ are not available prior to a time point $\Delta \geq 0$ and otherwise given by an unimodal function of time with $u(t) > 0 \forall t \geq \Delta$ representing the availability cycle of returns. Further, the return function has a maximum t_u^{max} and is continuously differentiable for $t > \Delta$.

A.3 There exists at most a single intersection point of demand and return functions $t_I \geq \max\{t_d^{max}, \Delta\}$ for which it holds $u(t) < d(t)$ if $t < t_I$ and $u(t) > d(t)$ if $t > t_I$. If there is no such intersection because demand always exceeds the return rate, t_I is assumed to equal infinity.

Figure 1 illustrates the situation under consideration. Using Δ and t_I , one can distinguish between three regions. In Region I, which ends at Δ , no returns are accessible. Demand has to be filled completely by producing new items. Region II shows less returns than demand (excess demand) and Region III is characterized by excess returns and decreasing demand.

Regarding the environmental policy of the firm and the existence of a strategic inventory that keeps returns for a later use, the following capital investment projects are considered:

- (a) **Design for single use.** Products are designed in such a way that they can not be remanufactured. Thus, all returns have to be disposed of at costs c_w , which can be positive if actual payments are necessary, or negative if there exists a positive salvage value. The corresponding investment at $t = 0$ is $K_p^s > 0$. Direct unit production costs are $c_p^s > 0$.

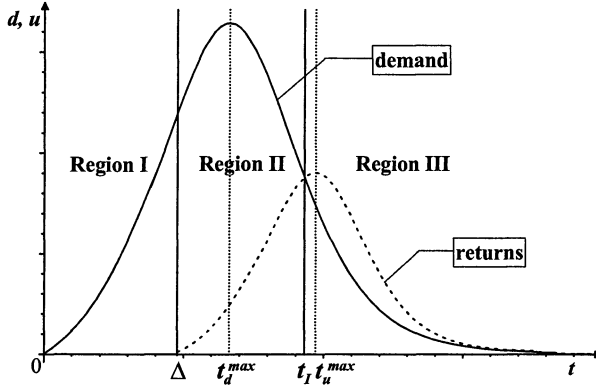


Fig. 1. A scenario satisfying assumptions A.1-A.3.

- (b) **Design for reuse.** Products are designed such that they can be made capable for another life time. Investment expenditures at $t = 0$ amount to $K_p^r > K_p^s$, producing each item would cost $c_p^r > c_p^s$. Both values exceed the above expenses for single use design because of additional requirements needed for remanufacturing. Used products can be remanufactured at costs $c_r > 0$ after introducing a remanufacturing process at time $t_r \geq \Delta$. This leads to a cash outflow of $K_r > 0$. A possibility to store returns is not considered. If returns exceed demand, they must be disposed of.
- (c) **Design for reuse with strategic inventory.** In addition to (b), it is possible to keep returns for later use in a recoverables inventory, e.g. at a time where remanufacturing is not yet possible, i.e. at $t < t_r$. The inventory level at t is denoted by $y_u(t)$. Out-of-pocket holding costs per item and time unit are given by $h_u > 0$.

The discount rate is denoted by α . Further assumptions are as follows: (1) A benefit must be realized if, instead of simultaneous production of a new item and disposal of the returned one, the latter is merely remanufactured, i.e. $c_p^r + c_w > c_r$. (2) It is not advantageous to delay the disposal of an unneeded returned item because saved interests on the required expenses are lower than out-of-pocket costs incurred by holding the item, i.e. $h_u > \alpha c_w$.

2.1 Valuation of Investment Project (a) - Design for Single Use

When assuming a single use product, the optimal dynamic policy is obviously to dispose of all returns immediately upon receipt, i.e. $w^*(t) = u(t)$. Demand is satisfied by synchronized production of new items ($p^*(t) = d(t)$). This leads to the following expression for the Net Present Value

$$NPV_a = K_p^s + \int_0^\infty e^{-\alpha t} [c_p^s d(t) + c_w u(t)] dt. \quad (1)$$

NPV_a can be considered as a benchmark against which the financial benefit of the other projects have to be compared.

2.2 Investment Project (b) - Design for Reuse

Dynamic Policy. The capital expenditure of K_r at time t_r subdivides the planning horizon into two parts. Before t_r , only production can be used to satisfy demand, and the same policy applies as for project (a). The optimal policy for $t \geq t_r$ is to remanufacture as many units as possible, to produce for excess demand, and to dispose of excess returns, yielding $p^*(t) = \max\{d(t) - u(t), 0\}$, $r^*(t) = \min\{d(t), u(t)\}$, $w^*(t) = \max\{u(t) - d(t), 0\} \forall t \geq t_r$.

The Net Present Value of project (b) NPV_b depends on the investment time t_r , and the following non-linear optimization problem needs to be solved

$$\begin{aligned} \min_{t_r} NPV_b(t_r) = & K_p^r + \int_0^{t_r} e^{-\alpha t} [c_p^r d(t) + c_w u(t)] dt + e^{-\alpha t_r} K_r \\ & + \int_{t_r}^{\infty} e^{-\alpha t} [c_p^r \max\{d(t) - u(t), 0\} + c_r \min\{u(t), d(t)\} + c_w \max\{u(t) - d(t), 0\}] dt. \end{aligned} \quad (2)$$

It can be shown that based on the assumptions A.1-A.3, $NPV_b(t_r)$ is strictly decreasing for $t_r < \Delta$ and at the limit, when t_r approaches infinity. Depending on the parameters, either there exists a single local minimum between Δ and t_I which is followed by a local maximum or the function is decreasing during the whole planning period $[0, \infty)$. The first derivative of $NPV_b(t_r)$ is a continuous function except for $t_r = \Delta$, given a jump discontinuity of the return function $u(t)$ at Δ .

Parameter Optimization. When choosing the investment time t_r , a trade-off has to be struck between the lower discounted value of the investment expenses K_r if the introduction of the remanufacturing process is postponed and a larger realized recovery cost advantage if it is placed earlier. Reconsidering the dynamic environment as introduced before, a finite optimal investment time t_r^* is located in Region II, and it must satisfy a break-even like condition.

Proposition 1. *If it exists, a finite investment time t_r^* must be located within the half open interval $[\Delta, \min\{t_u^{max}, t_I\})$, and one of the following situations must apply*

$$\begin{aligned} (i) \quad & u(t_r^*)(c_p^r + c_w - c_r) = \alpha K_r \quad \text{and} \quad \dot{u}(t_r^*) > 0 \quad \text{if} \quad t_r^* > \Delta, \\ (ii) \quad & u(t_r^*)(c_p^r + c_w - c_r) \geq \alpha K_r \quad \text{if} \quad t_r^* = \Delta. \end{aligned}$$

Proposition 1 states that at t_r^* , the current cost advantage of remanufacturing $u(t_r^*)(c_p^r + c_w - c_r)$ must at least earn interests on the investment, i.e. αK_r . If (i) equality holds, the remanufacturing rate must increase at t_r^* to start paying back the investment expenses. This is not possible later than either the time where returns reach their maximum t_u^{max} or the intersection time of demand and return rate t_I , because in both cases the remanufacturing rate must decrease thereafter. A special case (ii) may hold at time Δ , if there the return rate jumps. There exists no finite optimal investment time t_r^* , if the initial interest rate on the investment exceeds the maximum possible current cost advantage, i.e. $u(\min\{t_u^{max}, t_I\})(c_p^r + c_w - c_r) \leq \alpha K_r$.

So far, only local conditions have been considered. Of course, the investments required for the remanufacturing facility have to be paid off. By comparing the values of the objective of the finite candidate satisfying Proposition 1 with its limit as time approaches infinity, the following (global) condition for optimality of a finite investment time t_r^* results:

Proposition 2. *For the optimal investment time t_r^* it must hold that the total realized advantage of remanufacturing discounted to t_r^* at least equals the expenses needed for setting up the remanufacturing facility*

$$K_r \leq \int_{t_r^*}^{\infty} e^{-\alpha(t-t_r^*)} [c_p^r + c_w - c_r] \min\{d(t), u(t)\} dt. \quad (3)$$

Comparison with Investment Project (a). There exists no simple rule for determining the best of the two investment projects, but by comparing (optimal) Net Present Values it can be stated that project (b) is preferable to project (a) if the total discounted net advantage of remanufacturing $A_r^b = \int_{t_r^*}^{\infty} e^{-\alpha t} (c_p^r + c_w - c_r) \min\{d(t), u(t)\} dt - e^{-\alpha t_r^*} K_r$ exceeds the increase of the total discounted expenditures for the production process $D_p^b = (K_p^r - K_p^s) + \int_0^{\infty} e^{-\alpha t} [c_p^r - c_p^s] d(t) dt$.

2.3 Investment Project (c) - Design for Reuse with Strategic Inventory

Dynamic Policy. As an extension to project (b), returns can be stored in a recoverables inventory. Given a value for t_r and for the system's state at this time $y_u(t_r)$, the planning horizon can be subdivided into two parts. Prior to t_r , the question arises when to start collecting returns in order to achieve the desired stock. Therefore, to each value $y_u(t_r)$ a corresponding time point $t_e \leq t_r$ can be given where disposal stops and all returns are put to stock, being defined by $\int_{t_e}^{t_r} u(s) ds = y_u(t_r)$. Since stock-keeping can start earliest at Δ , the maximum possible quantity on stock at t_r is limited. Optimal decisions in the first part are given by $p^*(t) = d(t)$, $r^*(t) = 0$, $w^*(t) = u(t) \forall t < t_e$, and $p^*(t) = d(t)$, $r^*(t) = 0$, $w^*(t) = 0 \forall t_e \leq t < t_r$.

The solution of the second part is derived by using the optimal control framework presented in [4]. First, the recoverables inventory is depleted by filling excess demand from remanufacturing stored returns. This is completed at a time point $t_x \geq t_r$. Completion time t_x must not be larger than t_I because afterwards returns always exceed the demand rate. This gives another limiting condition for $y_u(t_r)$, i.e. $y_u(t_r) \leq \int_{t_r}^{t_I} (d(s) - u(s)) ds$. After t_x , the same policy is used as in investment project (b) because it is not useful to build up stock again, yielding the following optimal decisions in the second part $p^*(t) = 0$, $r^*(t) = d(t)$, $w^*(t) = 0 \forall t_r \leq t < t_x$, and $p^*(t) = \max\{d(t) - u(t), 0\}$, $r^*(t) = \min\{d(t), u(t)\}$, $w^*(t) = \max\{u(t) - d(t), 0\} \forall t \geq t_x$. Of course, this policy requires sufficient capacity and a high flexibility both in the production as well as in remanufacturing process.

In contrast to investment project (b), the optimal solution to policy class (c) also includes the starting time of collecting returns t_e^* . Restricting to finite solution candidates, we get the following optimization problem (4)-(7).

$$\begin{aligned} \min_{t_e, t_r} NPV_c(t_e, t_r, t_x) = & K_p^r + \int_0^{t_e} e^{-\alpha t} [c_p^r d(t) + c_w u(t)] dt \\ & + \int_{t_e}^{t_r} e^{-\alpha t} [c_p^r d(t) + h_u y_u(t)] dt + e^{-\alpha t_r} K_r + \int_{t_r}^{t_x} e^{-\alpha t} [c_r d(t) + h_u y_u(t)] dt \\ & + \int_{t_x}^{\infty} e^{-\alpha t} [c_p^r \max\{d(t) - u(t), 0\} + c_r \min\{u(t), d(t)\} + c_w \max\{u(t) - d(t), 0\}] dt \end{aligned} \quad (4)$$

$$\text{with } y_u(t; t_e, t_r, t_x) = \begin{cases} \int_{t_e}^t u(s) ds & \text{for } t \in [t_e, t_r] \\ \int_{t_e}^{t_r} u(s) ds - \int_{t_r}^t (d(s) - u(s)) ds & \text{for } t \in (t_r, t_x] \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and t_x being implicitly defined by a function $f(t_e, t_r, t_x)$

$$t_x : f(t_e, t_r, t_x) = \int_{t_e}^{t_r} u(s) ds - \int_{t_r}^{t_x} (d(s) - u(s)) ds = 0 \quad (6)$$

subject to the restrictions

$$\Delta \leq t_e, \quad t_e \leq t_r, \quad \int_{t_x}^{t_l} (d(s) - u(s)) ds \geq 0. \quad (7)$$

The objective function (4) incorporates all payments connected with the optimal policies in each of above distinguished regions. Function (5) is used to determine the inventory level and (6) gives an implicit definition of the time point t_x where the inventory is depleted. Constraints (7) limit the admissible set as described above and are needed in order to assure a meaningful solution.

Parameter Optimization. Due to the general assumptions on demand and return functions, objective (4) is not a (quasi-)convex function. Moreover, the admissible region is not even convex. Hence, conditions derived below by using standard methods of non-linear programming are only necessary for optimality. As there may exist several solution candidates, in order to find the optimal solution the objective values need to be compared.

In the following, by exploiting first order necessary conditions, four different types of solution candidates are distinguished. For ease of representation, a candidate is given by a triplet (t_e, t_r, t_x) , bearing in mind that t_x is a function of the other two time points. But firstly, a general condition regarding the optimal holding time $t_x^* - t_e^*$ is given. It must hold that $t_x^* - t_e^*$ does not exceed a maximal holding time τ , i.e.

$$t_x^* - t_e^* \leq \frac{1}{\alpha} \ln \left(\frac{\alpha(c_p^r - c_r) + h_u}{-\alpha c_w + h_u} \right) =: \tau. \quad (8)$$

This holding time τ comprises the same marginal criterion as known from [2] which balances the cost advantage of storing an otherwise disposed of returned item between t_e^* and t_x^* in order to replace production by remanufacturing at t_x^* and the required holding costs.

The following Propositions 3-6 present results for each of the cases.

Proposition 3 (Case (i) - interior solution). A triplet (t_e, t_r, t_x) with $\Delta < t_e < t_r < t_x < t_I$ is a solution candidate to problem (4)-(7) of Case (i), if $t_x - t_e = \tau$ and the following equation is satisfied

$$-e^{-\alpha t_r} [c_p^r - c_r] d(t_r) = h_u \int_{t_r}^{t_x} e^{-\alpha t} dt d(t_r) - e^{-\alpha t_x} [c_p^r - c_r] d(t_r) - \alpha e^{-\alpha t_r} K_r. \quad (9)$$

Since $t_x - t_e$ equals the maximal holding time τ , the decision maker must be indifferent between (1) disposing of a (marginal) return unit arriving at t_e , and producing a new one to meet demand at t_x or (2) holding this item until t_x when it is remanufactured to serve demand. Next, at t_r one needs to be indifferent between starting the remanufacturing process and thereby realizing the direct cost advantage of remanufacturing immediately or to postpone it which saves interests on the investment expenses. Then, a (marginal) demand $d(t_r)$ is served from producing new items and a thus saved (marginal) return is kept until t_x which results in holding costs and lowers the discounted value of the remanufacturing cost advantage. A Case 1 solution requires $t_I - \Delta > \tau$.

Proposition 4 (Case (ii) - complete use of interval $[\Delta, t_I]$). A triplet (t_e, t_r, t_x) with $\Delta = t_e < t_r < t_x = t_I$ is a solution candidate to problem (4)-(7) of Case (ii), if the following conditions are satisfied

$$e^{-\alpha \Delta} c_w d(t_r) \geq h_u \int_{\Delta}^{t_r} e^{-\alpha t} dt d(t_r) - e^{-\alpha t_r} [c_p^r - c_r] d(t_r) + \alpha e^{-\alpha t_r} K_r, \quad (10)$$

$$-e^{-\alpha t_r} [c_p^r - c_r] d(t_r) \geq h_u \int_{t_r}^{t_I} e^{-\alpha t} dt d(t_r) - e^{-\alpha t_I} [c_p^r - c_r] d(t_r) - \alpha e^{-\alpha t_r} K_r. \quad (11)$$

Inequation (10) implies that it would be preferable to put additional returns in stock at t_e for use at t_r by simultaneously lowering t_e and t_r , even at the cost of an earlier investment. But this is not possible because $\Delta = t_e$. Likewise, using (11), the value of the objective could be lowered by postponing investment time t_r . This is also forbidden because we would need to increase $t_x = t_I$ which again is not possible. A Case (ii) candidate only may exist, if $t_I - \Delta \leq \tau$. Therefore, it is not possible to have a situation where we could obtain solution candidates in both Case (i) and Case (ii).

Proposition 5 (Case (iii) - availability of returns is binding restriction). A triplet (t_e, t_r, t_x) with $\Delta = t_e < t_r < t_x < t_I$ is a solution candidate to problem (4)-(7) of Case (iii), if $t_x - \Delta \leq \tau$ and

$$-e^{-\alpha t_r} [c_p^r - c_r] d(t_r) = h_u \int_{t_r}^{t_x} e^{-\alpha t} dt d(t_r) - e^{-\alpha t_x} [c_p^r - c_r] d(t_r) - \alpha e^{-\alpha t_r} K_r. \quad (12)$$

As before, the maximal holding time is not yet reached. But, in contrast to Case (ii), from (12) we are indifferent regarding the postponement of t_r . Placing it earlier is also not possible, because t_e is fixed to Δ .

Proposition 6 (Case (iv) - availability of excess demand is binding restriction). A triplet (t_e, t_r, t_x) with $\Delta < t_e < t_r < t_x = t_I$ is a solution candidate to problem (4)-(7) of Case (iv), if $t_I - t_e \leq \tau$ and

$$e^{-\alpha t_e} c_w d(t_r) = h_u \int_{t_e}^{t_r} e^{-\alpha t} dt d(t_r) - e^{-\alpha t_r} [c_p^r - c_r] d(t_r) + \alpha e^{-\alpha t_r} K_r. \quad (13)$$

In a situation where t_x is fixed to t_I , choosing t_r requires indifference between disposing a (marginal) returned item at t_e or using it to lower t_r which in turn causes an increase in associated holding and interest expenses due to sooner investment but it also replaces production by remanufacturing at t_r .

Comparison with Investment Project (b). Since investment project (c) is a generalization of (b) which uses a strategic inventory to maximize the benefit from replacing production by remanufacturing, it generally leads to a lower Net Present Value. Another interesting question is how the possibility to hold returns for later use affects investment time t_r . Unfortunately, there is no unequivocal answer. An aspect that allows for postponing the investment time is that it no longer has a direct effect on the remanufacturability of returns since these also can be put to stock and remanufactured later. Other aspects make it possible to start remanufacturing earlier, e.g. a higher direct cost advantage of remanufacturing can be realized at t_r because demand is sourced completely from remanufacturing returns. For a detailed treatment of the planning problems discussed in this paper see [3].

3 Conclusions

In this paper we presented a framework to solve the technology choice problem with respect to the environmental policy in the presence of a product life cycle and an availability cycle for returns. Since we analyzed a quite simplistic situation, a number of possibilities exist for further research. A more complex demand/return situation can be solved by using general results for controlling the product recovery system as been shown in [4]. Capacity aspects did not play a role in our discussion. We assumed that the demand development does not depend on the chosen technology. Marketing aspects like consumer awareness towards environmental conscious products are neglected. Further, competition both on demand as well on the return side are not considered.

References

1. Guide Jr., V. D. R., Jayaraman, V., Srivastava, R., Benton, W. C. (2000) Supply-chain management for recoverable manufacturing systems. *Interfaces* 30:125–142
2. Kiesmüller, G. P., Minner, S., Kleber R. (2004) Managing dynamic product recovery: an optimal control perspective. In: Dekker, R., Inderfurth, K., Van Wassenhove, L., Fleischmann, M. (Eds.): *Quantitative Approaches to Reverse Logistics*. Springer, Berlin Heidelberg New York
3. Kleber R. (2004) A dynamic model for choosing the optimal technology in the context of reverse logistics. Working Paper, Otto-von-Guericke University Magdeburg
4. Minner, S., Kleber, R. (2001) Optimal control of production and remanufacturing in a simple recovery model with linear cost functions. *OR Spektrum* 23:3–24

Deriving Inventory-Control Policies for Periodic Review with Genetic Programming

Peer Kleinau¹, Ulrich W. Thonemann²

1 Münster University, Germany e-mail: peer.kleinau@gmx.de

2 Münster University, Germany e-mail: Ulrich.Thonemann@uni-muenster.de

Abstract. In Germany alone, inventories are estimated to be worth of more than 500 billion €. To manage these inventories, numerous inventory-control policies have been developed in the last decades. These inventory-control policies are typically derived analytically, which is often complicated and time consuming. For many relevant settings, such as complex multi-echelon models, there exist no closed-form formulae to describe the optimal solution. Optimal solutions for those problems are determined by complex algorithms that require several iteration steps. In this paper, we present an alternative approach to derive optimal or near-optimal inventory-control policies that are based on Genetic Programming (GP). GP is an algorithm related to Genetic Algorithms. It applies the principles of natural evolution to solve optimization problems. In this paper, we show how closed-form heuristics for a common inventory-control setting with periodic review can be found with GP. The advantage of GP is that inventory-control policies can be derived empirically without solving complex mathematical models.

Keywords Inventory - Genetic Programming

1 Introduction

In Germany, inventories are estimated to be worth of more than 500 billion €. In the last decades, a lot of efforts have been made to reduce these inventories. Today, various inventory-control policies are applied that are typically derived analytically, which is often complicated and time consuming. Moreover for many analytical inventory-control models, there exist no closed-form formulae to describe the optimal solution. In that cases, heuristics are applied that often fail to find optimal solutions or complex algorithms are used that require several calculation steps. With Genetic Programming (GP) however, inventory-control policies can be derived in a simple manner. In this paper, we show how closed-form heuristics for a common inventory-control setting can be found with GP. GP is applied to find closed-form formulae that provide good approximations for optimal solutions of the model. These closed-form formulae can be computed in a fraction of the time needed by exact methods. For a well-known setting, we demonstrate the capability of GP to provide good results. The remainder of this paper is divided into three sections. In the next section, we introduce GP. Then, we

apply GP to two common inventory-control problems: the newsvendor-model and a single-echelon (R,T)-model. In the last section, we sum up the results and conclude.

2 Genetic Programming

GP applies the principles of natural evolution to optimization problems [4]. It starts with an initial generation of artificial individuals that are created randomly. Then, a fitness value is assigned to each individual to describe quantitatively how well the individual masters its task. Based on these fitness values, the next generation is created. GP represents the individuals of each generation as trees. Each tree represents one solution to the underlying problem. In our application a tree represents a solution to an inventory-control policy. The generic GP algorithm consists of the following four steps [5]:

In the first step of the algorithm, the trees of the initial generation are created randomly. Each tree consists of nodes from a function set F and a terminal set T that are defined before the start of the algorithm. F contains the functions that can be used to construct the trees, such as $+$, $-$, $*$ and $/$. T contains the input parameters and state variables of the problem, such as demand rates and inventory levels. T also contains a placeholder that represents a real number. If this placeholder is used in the tree, a random number is generated and placed at the corresponding location of the tree. Figure 1 shows the construction of a tree. The first node of a tree is always a function; subsequent nodes are selected from the union $F \cup T$. In Figure 1 the first node of the tree is the multiplication function. Since this function requires two arguments, two arcs are attached to the node. Additional nodes are then randomly chosen from $F \cup T$ and attached to the arcs. For the left arc, the terminal L is selected; for the right arc, the terminal λ is selected. Because terminals have no arguments, the construction of the tree is completed. The tree represents the formula $L \cdot \lambda$. Since trees are constructed by choosing functions and terminals randomly, a wide variety of trees can be created.

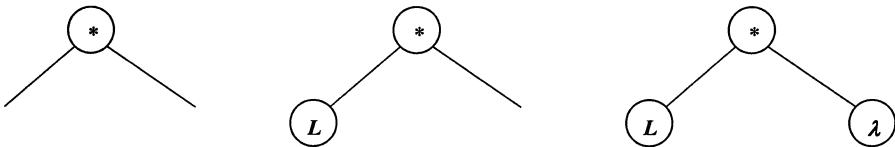


Fig. 1. Construction of a tree of the initial generation

In the second step of the algorithm, a fitness value is assigned to each tree. Trees that solve the underlying problem well receive a higher fitness value than trees that solve the underlying problem poorly. In our application each tree represents an inventory-control policy. To evaluate the performance of the policy, we apply it to

control an inventory systems and compute the average cost that is achieved under this policy.

In the third step of the algorithm, the trees of the next generation are created by applying the genetic operators crossover, mutation, and reproduction to the trees of the current population. The probability that a tree is selected for a genetic operation depends on its fitness value. Trees with a high fitness value have a higher probability of being selected than trees with a low fitness value. Thus, the selection favors trees that provide a good solution to the problem.

In the fourth step of the algorithm, it is checked if a certain number of generations has been created. If a pre-defined limit is reached, the algorithm terminates and the best tree found during the run is reported as the result of the run. Otherwise steps 2 to 4 are repeated.

3 Applying Genetic Programming to Inventory Control

3.1 Description of Inventory-Control Models

Companies hold inventories, among other reasons, to hedge against demand uncertainty, and to realize economies of scale. Holding inventory results in inventory-holding costs. The inventory-holding cost rate h is used to represent these costs. Although inventories result in costs, they can be beneficial. If an order causes setup cost k , it is often useful not to order a single unit, but batches of several units. Order batching increases inventory-holding cost, but decreases order-setup cost. The optimal batch size has to be determined as a trade-off between those two cost factors. With deterministic demand, inventories cover future demand exactly. With stochastic demand, future demands cannot be forecasted precisely. We assume that if customers' demand cannot be satisfied, customers are willing to wait for unsatisfied demand, yielding backlogging costs. The backlogging cost rate p exceeds usually the inventory-holding cost rate h . As a consequence, companies hold safety stocks to meet stochastic demand with low backlogging costs. High safety stocks lead to low backlogging costs, but to high inventory-holding cost. Therefore, the optimal safety stock has to be determined.

3.2 Newsvendor-Model

To analyze this situation, various models exist [8]. One basic model is the newsvendor-model [1]. Demand is stochastic, and orders arrive immediately, i.e., the lead time is zero. At the beginning of each period, the inventory level (inventory on-hand) is reviewed and an order is placed to raise the inventory level

up to S . At the end of each period, inventory-holding cost and backorder-penalty cost are charged.

3.3 (R, T) -Model

One of the most common inventory-control models for periodic review follows a order-up-to policy under periodic review, and is referred to as the (R, T) -model. Demand is stochastic, and orders arrive after a lead time of L . In our setting, demand follows a Poisson process with mean λ . The inventory position (inventory on-hand plus on-order minus backorder) is reviewed every T periods and each period an order is placed to raise the inventory position up to R . The optimal values for T and for R have to be determined [2]. There exist algorithms to compute optimal values for R and T , but there exists no closed-form formulae. The optimal solution algorithms rely on iterative methods, such as the Newton's method [2] or other approaches for convex optimization [6].

3.4 Structure of Analysis

In this section, we will show how GP can be used to generate optimal or near-optimal solutions for two commonly used inventory-control models, the newsvendor-model and the (R, T) -model. We select these models, because they belong to the most frequently used models. First, we only search for one parameter. We analyze whether GP is capable to rediscover the optimal solution of the newsvendor-model. Next, we simplify the (R, T) -model. We do not search for the optimal values for both R and T simultaneously, but we optimize R for a given T . The average cost of the solutions found by GP are calculated using analytical expressions for the cost function of the (R, T) -model. For this setting, Hadley and Whitin [2] derived a approximate cost function, where a closed-form solution to compute the optimal value of R exist, for a given T . We compare the solutions found by GP with this formula. The results of our experiments indicate that GP is able to find the optimal policy of the newsvendor-model and a good heuristic for the simplified (R, T) -model. Finally, we use GP to search for two functions, representing R and T . We compare the solutions found with a common heuristic and a numerical approach.

3.5 Search for the Optimal Solution of the Newsvendor-Model

We use the function set $F = \{+, -, *, /, ^, \sqrt{}, I\}$ and the terminal set $T = \{h, p, \mathcal{R}\}$. $^$ is the power function. I is the inverse cumulative Poisson distribution. \mathcal{R} is a placeholder for randomly generated numerical values in the range $[0 .. 10]$.

The evaluation of an individual is based on its average performance over a number of training cases. To evaluate an individual, we compute its cost by applying

$$TC(S) = h \int_0^S (S - x)f(x)dx + p \int_S^{\infty} (x - S)f(x)dx \quad (1)$$

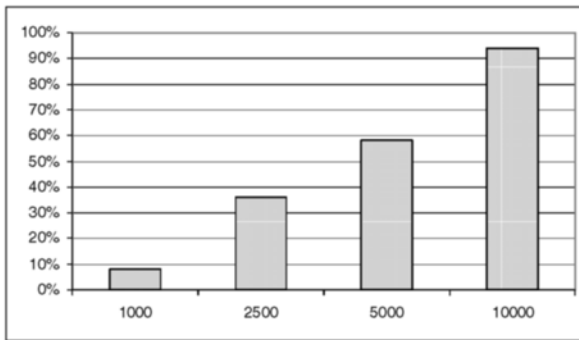
To train the algorithm, we apply each individual to a set of 100 training cases that are based on randomly generated combinations of the parameters λ, h, p , with λ [0.5 .. 5], h [1 .. 50], p [0.1 h .. 25 h]. For each parameter combination, the optimal value of S^* can be calculated as follows

$$I\left(\frac{p}{p+h}\right) = S^* \quad (2)$$

For each training case, we can compute the cost of the optimal solution. Then, we compare the optimal cost with the cost of the individual for each training case and sum up the deviations. The higher the cost of the individual in comparison to the optimal cost, the lower the fitness of the individual and therefore, the change to be selected for genetic operations.

We conduct 50 runs of 30 generations each, with population sizes of 1,000, 2,500, 5,000, and 10,000 individuals to analyse the capability of GP to find this formula. The percentage of hits, i.e., the percentage of runs where GP finds the optimal solution, increases with an increase of the population size. In all experiments, we apply a crossover probability of 90%, a mutation probability of 5%, and a reproduction probability of 5%. Figure 2 reports the percentage of runs that find the optimal solution.

Percentage of hits



Population size

Fig. 2. Results of search for S^*

3.6 Search for R

We use the function set $F = \{+, -, *, /, ^, \sqrt{}, I, CI\}$ and the terminal set $T = \{h, p, k, T, R\}$. CI is the complementary of the inverse cumulative Poisson distribution. The evaluation of an individual is based on its average performance over a number of training cases. To evaluate an individual, we compute its cost by an adoption of Rao's [6] exact cost function for a number of training cases.

$$TC(R, T) = \frac{k}{T} + \frac{1}{T} \int_L^{L+T} G(R, t) dt, \text{ with} \quad (3)$$

$$G(R, t) = hE \left[\frac{X(t) - R}{h} \right]^+ + pE \left[\frac{X(t) - R}{h} \right]^+ \quad (4)$$

$X(t)$ is the cumulative demand distribution. To train the algorithm, we apply each individual to a set of 100 training cases that are based on randomly generated combinations of the parameters λ, L, T, h, p, k , with the ranges λ [0.5 .. 5], L [0 .. 3], h [1 .. 50], p [0.1 h .. 25 h], k [1 .. 1000].

For each training case, we can compute the cost of the optimal solution. Then, we compare the optimal cost with the cost of the individual for each training case and sum up the deviations. The higher the cost of the individual in comparison to the optimal cost, the lower the fitness of the individual and therefore, the change to be selected for further generations.

After 724 generations with a population size of 10.000, GP found a solution with average cost 0.6% above the optimal solution for the training cases. R can be computed by solving

$$CI \left(\frac{\sqrt{9.5I}}{\frac{p}{h} + \sqrt{9.5I}} \right) \quad (5)$$

To test the performance of the solution found by GP, we compare its cost with the cost of the solution found by Hadley and Whitin [2] on a basis of 1000 test cases. Hadley and Whitin developed a approximate cost function, where the optimal R can be computed for a given T by solving

$$CI \left(\frac{hT}{p} \right) \quad (6)$$

Based on a test bed of 1000 test cases, the average cost of the GP solution was 0.8% above optimality, whereas the average cost of the solution of Hadley and Whitin was 3.7% above optimality.

3.7 Search for R and T

We next are to search for both R and T simultaneously. Each GP tree consists of two subtrees. One subtree represents R , the other one T . The parameters λ and L are added to the terminal set.

We apply each individual to 100 training cases that are based on randomly generated combinations of the parameters λ [0.5 .. 5], L [0 .. 3], h [1 .. 50], p [0.1 h .. 25 h], k [1 .. 1000]. In this phase, T is not predefined, but has to be determined by GP. We compute the optimal solutions and compare their cost with the cost of the inventory-control policies created by GP. After more than 1000 generations, GP found the following solution, with average cost 7.37% above optimality.

$$CI \left(\frac{\lambda}{\frac{p}{h} + \lambda} \right) \text{ and } T = \sqrt{\frac{2k + h\lambda}{h\lambda}} - n + 7.18 \quad (7)$$

To test, whether GP has found a general solution, we compare it with solutions found by other approaches. We selected two methods: two heuristic approaches and a numerical, optimal one.

The first heuristic is suggested by Tempelmeier [7] to compute T . For this T , R is computed using Hadley and Whitin's formula [2]. T is calculated using the EOQ formula [3].

$$T = \frac{EOQ}{\lambda} = \frac{\sqrt{\frac{2k\lambda}{h}}}{\lambda} = \sqrt{\frac{2k}{h\lambda}} \quad (8)$$

The second heuristic computes T using the EOQ formula. Then, R is computed by applying the optimal solution of the newsvendor-model. The optimal approach uses Newton's method to calculate the optimal values for R and T .

We compare the performance of the heuristics using a test bed of 1000 cases. For this test bed, the optimal solutions are calculated with Newton's method. The Genetic Programming solution was on average 4.47% above optimality, whereas the heuristic of Tempelmeier was on average 15.40% above optimality. The heuristic based on the solution to the modified newsvendor model was on average 47.47% above optimality. Based on this test bed of 1000 cases, we conclude that Genetic Programming finds a solution in closed-form that is superior to commonly used heuristics for the used ranges of parameters.

4 Conclusions

Inventory-control is a broad field of research. Numerous models have been developed to optimize inventory systems. For the simple ones, such as the newsvendor-model, optimal solutions have been found, for the complex ones, heuristics have been developed. In this paper, we show how closed-form heuristics can be found with Genetic Programming. For a simple model, the newsvendor-model, we rediscover the optimal solution. For the (R,T)-model, we find heuristics that provide a good solution quality with a low complexity. As a result of our work, we can conclude that Genetic Programming is a powerful tool for deriving inventory-control policies. However, there is no guarantee that inventory-control policies found by GP perform decently for parameter combinations outside of the tested ranges.

References

1. Arrow, K.A., Harris, T.E., Marschak, T. 1951. Optimal inventory Policy. *Econometrica* 19, 250-272.
2. Hadley, G., Whitin, T. 1963. *Analysis of Inventory Systems*. Englewood Cliffs, NY, Prentice-Hall.
3. Harris, F. W. 1915. *Operations and Costs*. Factory Management Series, Chicago.
4. Koza, J. R. 1992. *Genetic Programming – On the programming of computers by means of natural selection*. MIT Press, Cambridge.
5. Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A. 1999. *Genetic Programming III – Darwinian Invention and Problem Solving*. Kaufmann, San Francisco.
6. Rao, U. S. 2003. Properties of the Periodic Review (R,T) Inventory Control Policy for Stationary. Stochastic Demand, *Manufacturing & Service Operations Management* 5(1), 37-53.
7. Tempelmeier, H. 2003. *Material-Logistik: Modelle für die Produktionsplanung und –steuerung und das Supply Chain Management*. Springer, Berlin.
8. Zipkin, P. H. 2000. *Foundations of inventory management*. McGraw-Hill, Boston.

Dynamic Multi-Commodity Facility Location: A Mathematical Modeling Framework for Strategic Supply Chain Planning

M. T. Melo¹, S. Nickel^{1,2}, and F. Saldanha da Gama³

¹ Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), D 67663 Kaiserslautern, Germany

² Chair of Operations Research and Logistics, Saarland University, D 66041 Saarbrücken, Germany

³ Department of Statistics and Operations Research, University of Lisbon, P 1749-016 Lisbon, Portugal

Abstract. We propose a mathematical modeling framework for supply chain network design which captures many aspects of practical problems that have not received adequate attention in the literature. The aspects considered include: dynamic planning horizon, generic supply chain structure, inventory and distribution opportunities for goods, facility configuration, budget constraints, and storage limitations. Moreover, the gradual relocation of facilities over the planning horizon is considered. A generic mathematical programming model is described in detail and several extensions are discussed.

1 Introduction

The design of supply chain networks is a complex decision making process. Typical decisions involve determining the location, size and the number of new facilities to operate as well as the flow of goods through the supply chain network to satisfy known demands. Facility location and configuration of production–distribution networks have been studied for many years (see Bender *et al.* [1] for a review of models, company-specific case studies, and decision support systems). Important issues such as the external supply of commodities, inventory opportunities for goods, storage limitations, availability of capital for investments, and relocation, expansion or reduction of capacities have been treated individually by several authors. However, there is evidence that companies wish them all to be explicitly included in the design of their supply chain networks (see e.g., Bender *et al.* [1] and Kalcsics *et al.* [2]). Clearly, network design is strongly affected by the simultaneous consideration of these and other practical needs. One observes a lack of reasonably simple, yet comprehensive, models which illustrate the effect of such factors on network configuration decisions. This paper attempts to fill this gap. Our main contribution is to provide a mathematical modeling framework for assisting decision-makers in the design of their supply chain networks. In

the next section we describe the problem at stake. A mixed integer linear programming (MIP) formulation is introduced in Section 3. Finally, Section 4 briefly discusses several model extensions and directions for future research.

2 Dynamic Facility Relocation

We assume that a company is considering gradually relocating part or all of the capacity of some of its existing facilities to new locations during a certain time horizon. Prior to the planning project, a set of candidate sites has been selected where new facilities can be established. Both the existing capacities as well as the capacities of the prospective sites are known in advance. To illustrate the conditions under which capacity may be transferred from existing locations to new sites, consider the situation depicted in Figure 1. At the end of a given period, say $t - 1$, the company operates facilities i_1 ,

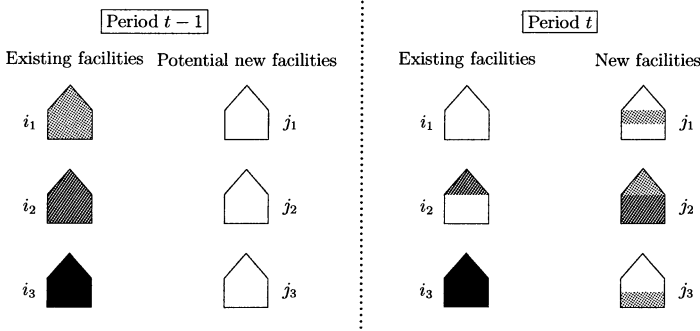


Fig. 1. The effect of capacity relocation

i_2 and i_3 . Potential sites for setting up new facilities are denoted by j_1 , j_2 and j_3 . It is assumed that if capacity is to be shifted then this will occur at the beginning of period t , and will have a relatively short duration compared to the period length. On the right-hand side of Figure 1 a possible scenario for capacity shifting is displayed. At the beginning of period t new facilities will be operating in all three potential sites. The capacity of the existing facility i_1 is totally distributed among the sites j_1 , j_2 , and j_3 . As a result, facility i_1 will neither operate in period t nor in any subsequent period. Thus, we assume that the facility is closed at the end of period $t - 1$, and its total capacity is moved to the new sites at the beginning of period t . Part of the capacity of the existing facility i_2 is transferred to the new site j_2 . Facility i_2 remains in operation but with reduced capacity. Note that the new facility j_2 has attained its maximum capacity, while facilities j_1 and j_3 can still have their capacity extended in later periods. Capacities transferred to the new facilities cannot be removed in later periods, that is, capacity reductions are

only allowed at existing facilities. Finally, the existing facility i_3 retains all its capacity.

Relocation decisions are constrained by budget limitations. We assume that in each time period a given budget is available for investing not only in capacity transfers, but also in the setup of new facilities and shutdown of existing facilities. Relocation costs incurred by capacity shifts are assumed to depend on the amount moved from an existing facility to a new site, and account, for example, for workforce and equipment transfers. Since the setup of a new facility is usually a time-consuming process, we assume that it takes place in the period immediately preceding the start-up of operations. Hence, if a new facility starts operating in some period t , fixed costs are charged with respect to its setup in period $t - 1$. On the other hand, when an existing facility ceases operating at the end of some period t , we assume that shutdown costs are charged in the following period. Any capital available in a period, but not invested then, is subject to an interest rate and the returned value can be used in subsequent periods.

Using the available investment capital in each period and considering the capacity limitations, our problem consists in determining the time periods during which the capacity of existing facilities should be reduced and when the transition to newly established facilities should be made. Capacity shifts must be carried out in such a way that supply chain activities are not disrupted. The latter include the provision of materials from external suppliers, the transportation of goods through the network and the storage of products at facilities. These activities are to be planned so as to minimize system-wide costs given that demands must be satisfied. Finally, in contrast to many well-known location models, we neither impose an echelon structure on the supply chain network nor restrict the distribution channels for shipping the goods. In other words, any type of facility is supported by our approach and commodities can be transported between any pair of facilities.

3 Problem Formulation

Index sets

\mathcal{L} : set of facilities

\mathcal{S} : set of *selectable* facilities, $\mathcal{S} \subset \mathcal{L}$

\mathcal{S}^c : set of *selectable existing* facilities, $\mathcal{S}^c \subset \mathcal{S}$

\mathcal{S}^o : set of potential sites for establishing *new* facilities, $\mathcal{S}^o \subset \mathcal{S}$

\mathcal{P} : set of product types

\mathcal{T} : set of periods with $|\mathcal{T}| = n$

The set \mathcal{L} contains all types of facilities. These are categorized in so-called *selectable* and *non-selectable* facilities. Selectable facilities form the set \mathcal{S} and include existing facilities (\mathcal{S}^c) as well as potential sites for establishing new

facilities (S^o). At the beginning of the planning horizon, all the facilities in the set S^c are operating. Afterwards, capacity can be shifted from these facilities to new facilities located at the sites in S^o . Note that $S^c \cap S^o = \emptyset$ and $S^c \cup S^o = S$. The second category of facilities, the so-called non-selectable group, forms the set $\mathcal{L} \setminus S$ and includes all facilities that exist at the beginning of the planning project and which will remain in operation. Examples of such facilities include plants and warehouses that should continue supporting supply chain activities, that is, are not subject to relocation decisions. Non-selectable facilities may also have demand requirements, that is, they may correspond to customers.

Costs

- $PC_{\ell,p}^t$: variable cost of purchasing one unit of product $p \in \mathcal{P}$ from an external supplier by facility $\ell \in \mathcal{L}$ in period $t \in \mathcal{T}$
- $TC_{\ell,\ell',p}^t$: variable cost of shipping one unit of product $p \in \mathcal{P}$ from facility $\ell \in \mathcal{L}$ to facility $\ell' \in \mathcal{L}$ ($\ell \neq \ell'$) in period $t \in \mathcal{T}$
- $IC_{\ell,p}^t$: variable inventory carrying cost per unit on hand of product $p \in \mathcal{P}$ in facility $\ell \in \mathcal{L}$ at the end of period $t \in \mathcal{T}$
- $MC_{i,j}^t$: unit variable cost of moving capacity from the existing facility $i \in S^c$ to a new facility established at site $j \in S^o$ at the beginning of period $t \in \mathcal{T} \setminus \{1\}$
- OC_{ℓ}^t : fixed cost of operating facility $\ell \in \mathcal{L}$ in period $t \in \mathcal{T}$
- SC_i^t : fixed cost charged in period $t \in \mathcal{T} \setminus \{1\}$ for having shut down the existing facility $i \in S^c$ at the end of period $t - 1$
- FC_j^t : fixed setup cost charged in period $t \in \mathcal{T} \setminus \{n\}$ when a new facility established at site $j \in S^o$ starts its operation at the beginning of period $t + 1$

Parameters

- \overline{K}_{ℓ}^t : maximum capacity of facility $\ell \in \mathcal{L}$ in period $t \in \mathcal{T}$
- \underline{K}_{ℓ}^t : minimum required throughput at the selectable facility $\ell \in S$ in period $t \in \mathcal{T}$
- $\mu_{\ell,p}$: unit capacity consumption factor of product $p \in \mathcal{P}$ at facility $\ell \in \mathcal{L}$
- $H_{\ell,p}$: stock of product $p \in \mathcal{P}$ at facility $\ell \in \mathcal{L}$ at the beginning of the planning horizon (observe that $H_{\ell,p} = 0$ for every $\ell \in S^o$)
- $D_{\ell,p}^t$: external demand of product $p \in \mathcal{P}$ at facility $\ell \in \mathcal{L}$ in period $t \in \mathcal{T}$
- α^t : unit return factor on capital not invested in period $t \in \mathcal{T} \setminus \{n\}$, that is, $\alpha^t = 1 + \beta^t/100$ with β^t denoting the interest rate in period t
- B^t : available budget in period $t \in \mathcal{T}$

Since each existing selectable facility may have its capacity transferred to one or more new facilities, it is assumed that its maximum capacity is non-increasing during the planning horizon, that is, $\bar{K}_i^t \geq \bar{K}_i^{t+1}$ for every $i \in \mathcal{S}^c$ and $t \in \mathcal{T} \setminus \{n\}$. Without loss of generality, it is assumed that \bar{K}_i^1 denotes the actual size of facility $i \in \mathcal{S}^c$ at the beginning of the planning horizon. The above condition permits to impose capacity transfers in specific periods or even complete shutdowns. Similarly, potential new facilities have non-decreasing capacities, that is, $\bar{K}_j^t \leq \bar{K}_j^{t+1}$ for every $j \in \mathcal{S}^o$ and $t \in \mathcal{T} \setminus \{n\}$. Clearly, at the beginning of the planning project we have $\bar{K}_j^1 = 0$ for every new site $j \in \mathcal{S}^o$.

Decision variables

- $b_{\ell,p}^t$ = amount of product $p \in \mathcal{P}$ purchased from an outside supplier by facility $\ell \in \mathcal{L}$ in period $t \in \mathcal{T}$
- $x_{\ell,\ell',p}^t$ = amount of product $p \in \mathcal{P}$ shipped from facility $\ell \in \mathcal{L}$ to facility $\ell' \in \mathcal{L}$ ($\ell \neq \ell'$) in period $t \in \mathcal{T}$
- $y_{\ell,p}^t$ = amount of product $p \in \mathcal{P}$ held in stock in facility $\ell \in \mathcal{L}$ at the end of period $t \in \mathcal{T} \cup \{0\}$ (observe that $y_{\ell,p}^0 = H_{\ell,p}$)
- $z_{i,j}^t$ = amount of capacity shifted from the existing facility $i \in \mathcal{S}^c$ to a newly established facility at site $j \in \mathcal{S}^o$, at the beginning of period $t \in \mathcal{T}$
- ξ^t = capital not invested in period $t \in \mathcal{T}$
- $\delta_\ell^t = \begin{cases} 1 & \text{if the selectable facility } \ell \in \mathcal{S} \text{ is operated during period } t \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases}$

In view of the assumptions made on the time points for paying fixed facility costs, it follows that a new facility can never operate in the first period since that would force the company to invest in its setup before the beginning of the planning horizon. Similarly, an existing facility cannot be closed at the end of the last period since the corresponding fixed shutdown costs would be charged in a period beyond the planning horizon. Hence, $z_{i,j}^1 = 0$ for $i \in \mathcal{S}^c$ and $j \in \mathcal{S}^o$, $\delta_i^1 = 1$ for $i \in \mathcal{S}^c$, and $\delta_j^1 = 0$ for $j \in \mathcal{S}^o$.

Assuming that all parameters are non-negative our MIP formulation is as follows.

$$\begin{aligned} \text{MIN} \quad & \sum_{t \in \mathcal{T}} \sum_{\ell \in \mathcal{L}} \sum_{p \in \mathcal{P}} PC_{\ell,p}^t b_{\ell,p}^t + \sum_{t \in \mathcal{T}} \sum_{\ell \in \mathcal{L}} \sum_{\ell' \in \mathcal{L} \setminus \{\ell\}} \sum_{p \in \mathcal{P}} TC_{\ell,\ell',p}^t x_{\ell,\ell',p}^t \\ & + \sum_{t \in \mathcal{T}} \sum_{\ell \in \mathcal{L}} \sum_{p \in \mathcal{P}} IC_{\ell,p}^t y_{\ell,p}^t + \sum_{t \in \mathcal{T}} \sum_{\ell \in \mathcal{S}} OC_\ell^t \delta_\ell^t + \sum_{t \in \mathcal{T}} \sum_{\ell \in \mathcal{L} \setminus \mathcal{S}} OC_\ell^t \end{aligned} \quad (1)$$

s. t.

$$b_{\ell,p}^t + \sum_{\ell' \in \mathcal{L} \setminus \{\ell\}} x_{\ell',\ell,p}^t + y_{\ell,p}^{t-1} =$$

$$D_{\ell,p}^t + \sum_{\ell' \in \mathcal{L} \setminus \{\ell\}} x_{\ell,\ell',p}^t + y_{\ell,p}^t, \quad \ell \in \mathcal{L}, p \in \mathcal{P}, t \in \mathcal{T} \quad (2)$$

$$\bar{K}_i^1 - \sum_{\tau=1}^t \sum_{j \in \mathcal{S}^o} z_{i,j}^\tau \leq \bar{K}_i^t \delta_i^t, \quad i \in \mathcal{S}^c, t \in \mathcal{T} \quad (3)$$

$$\sum_{\tau=1}^t \sum_{i \in \mathcal{S}^c} z_{i,j}^\tau \leq \bar{K}_j^t \delta_j^t, \quad j \in \mathcal{S}^o, t \in \mathcal{T} \quad (4)$$

$$\sum_{\tau=1}^t \sum_{j \in \mathcal{S}^o} z_{i,j}^\tau + \delta_i^t \epsilon \leq \bar{K}_i^1, \quad i \in \mathcal{S}^c, t \in \mathcal{T} \quad (5)$$

$$\begin{aligned} & \sum_{p \in \mathcal{P}} \mu_{i,p} \left(b_{i,p}^t + \sum_{\ell \in \mathcal{L} \setminus \{i\}} x_{\ell,i,p}^t + y_{i,p}^{t-1} \right) \\ & \leq \bar{K}_i^1 - \sum_{\tau=1}^t \sum_{j \in \mathcal{S}^o} z_{i,j}^\tau, \quad i \in \mathcal{S}^c, t \in \mathcal{T} \end{aligned} \quad (6)$$

$$\begin{aligned} & \sum_{p \in \mathcal{P}} \mu_{j,p} \left(b_{j,p}^t + \sum_{\ell \in \mathcal{L} \setminus \{j\}} x_{\ell,j,p}^t + y_{j,p}^{t-1} \right) \\ & \leq \sum_{\tau=1}^t \sum_{i \in \mathcal{S}^c} z_{i,j}^\tau, \quad j \in \mathcal{S}^o, t \in \mathcal{T} \end{aligned} \quad (7)$$

$$\sum_{p \in \mathcal{P}} \mu_{\ell,p} \left(b_{\ell,p}^t + \sum_{\ell' \in \mathcal{L} \setminus \{\ell\}} x_{\ell',\ell,p}^t + y_{\ell,p}^{t-1} \right) \leq \bar{K}_\ell^t, \quad \ell \in \mathcal{L} \setminus \mathcal{S}, t \in \mathcal{T} \quad (8)$$

$$\sum_{p \in \mathcal{P}} \mu_{\ell,p} \left(b_{\ell,p}^t + \sum_{\ell' \in \mathcal{L} \setminus \{\ell\}} x_{\ell',\ell,p}^t + y_{\ell,p}^{t-1} \right) \geq \underline{K}_\ell^t \delta_\ell^t, \quad \ell \in \mathcal{S}, t \in \mathcal{T} \quad (9)$$

$$\delta_i^t \geq \delta_i^{t+1}, \quad i \in \mathcal{S}^c, t \in \mathcal{T} \setminus \{n\} \quad (10)$$

$$\delta_j^t \leq \delta_j^{t+1}, \quad j \in \mathcal{S}^o, t \in \mathcal{T} \setminus \{n\} \quad (11)$$

$$\sum_{j \in \mathcal{S}^o} FC_j^1 \delta_j^2 + \xi^1 = B^1 \quad (12)$$

$$\begin{aligned} & \sum_{i \in \mathcal{S}^c} \sum_{j \in \mathcal{S}^o} MC_{i,j}^t z_{i,j}^t + \sum_{i \in \mathcal{S}^c} SC_i^t (\delta_i^{t-1} - \delta_i^t) + \sum_{j \in \mathcal{S}^o} FC_j^t (\delta_j^{t+1} - \delta_j^t) \\ & + \xi^t = B^t + \alpha^{t-1} \xi^{t-1}, \quad t \in \mathcal{T} \setminus \{1, n\} \end{aligned} \quad (13)$$

$$\begin{aligned} & \sum_{i \in \mathcal{S}^c} \sum_{j \in \mathcal{S}^o} MC_{i,j}^n z_{i,j}^n + \sum_{i \in \mathcal{S}^c} SC_i^n (\delta_i^{n-1} - \delta_i^n) + \xi^n \\ & = B^n + \alpha^{n-1} \xi^{n-1} \end{aligned} \quad (14)$$

$$b_{\ell,p}^t \geq 0, \quad y_{\ell,p}^t \geq 0, \quad x_{\ell,\ell',p}^t \geq 0, \quad \ell, \ell' \in \mathcal{L}, p \in \mathcal{P}, t \in \mathcal{T} \quad (15)$$

$$z_{i,j}^t \geq 0, \quad i \in \mathcal{S}^c, j \in \mathcal{S}^o, t \in \mathcal{T} \quad (16)$$

$$\xi^t \geq 0, \quad t \in \mathcal{T} \quad (17)$$

$$\delta_\ell^t \in \{0, 1\}, \quad \ell \in \mathcal{S}, t \in \mathcal{T} \quad (18)$$

In the above formulation, the objective is to operate the supply chain network at minimum cost as stated by (1). The costs to minimize include supply costs, transportation costs between facilities, inventory holding costs, and fixed facility operating costs.

Constraints (2) are the usual flow conservation conditions which must hold for each product, facility and period. Inequalities (3)–(5) ensure that feasible capacity relocations take place during the planning horizon. Note that constraints (3) also guarantee that only operating existing facilities can have their capacity moved to new facilities. Furthermore, constraints (4) also impose that by period t a new facility has been constructed at site j in order for a capacity relocation to take place. Constraints (5) (with ϵ a sufficiently small positive number) state that if the capacity of an existing facility has been completely transferred then the facility has to be closed. The combination of (3) and (5) ensures that if an existing facility does not operate in a given period then its entire capacity was removed in one of the previous periods. Moreover, by (5) no more capacity can be shifted out of such a facility than the one available at the beginning of the planning horizon. Inequalities (6)–(8) impose the capacity constraints. Observe that inequalities (6) also prevent any supply chain activities from taking place in existing facilities whose capacity has been totally relocated. Inequalities (9) state that it is only worth to operate a selectable facility if its throughput is above a meaningful level. Constraints (10) and (11) allow the configuration of each selectable facility to change at most once. Hence, if an existing facility is closed, it cannot be re-opened. Similarly, when a new facility is established it will remain in operation until the end of the planning horizon. Conditions (12)–(14) are budget constraints. In the first period, the allowed investments regard setting up new facilities that will start operating at the beginning of the second period (cf. (12)). In each one of the following periods $t \in \mathcal{T} \setminus \{1, n\}$, the available capital may cover capacity transfers, the costs incurred by closing existing facilities at the end of period $t - 1$, and setups of new facilities that start operating at the beginning of period $t + 1$ (cf. (13)). In the last period n , the allowed investments concern capacity transfers as well as shutdowns of facilities that have ceased operating at the end of period $n - 1$ (cf. (14)). Finally, constraints (15)–(18) represent non-negativity and integrality conditions.

The above formulation describes a large scale MIP problem. Using commercially available mathematical programming software, Melo *et al.* [3] have shown that problems of moderate size (with up to $|\mathcal{T}| = 5$, $|\mathcal{L}| = 185$, $|\mathcal{S}^c| = 10$, $|\mathcal{S}^o| = 20$, $|\mathcal{S}| = 30$, and $|\mathcal{P}| = 10$) can be solved optimally in less than 5 hours (using a Pentium III, 2.6 GHz, 2 GB RAM).

4 Extensions and Outlook

When a company anticipates a growth in its sales volumes, it will need to expand its supply chain network to respond to increasing demand patterns. Expansion plans may result in extending the capacity of existing facilities and/or establishing additional facilities. Although in our model the total capacity available at the beginning of the planning project does not change over the time horizon, it is relatively easy to extend it to an expansion scenario. We consider a *fictitious facility*, denoted by i_0 , which concentrates the total additional capacity, $\bar{K}_{i_0}^1$, required to cope with increasing demands. This facility has a similar status to that of the selectable facilities in \mathcal{S}^c , although it cannot satisfy any demands. Its capacity can be shifted to both existing and new facilities. As a result, either an existing facility will have its capacity expanded or it will be (partly) relocated to one or more new sites. Regarding the new facilities, these can receive capacity not only from existing (selectable) facilities, but also from the fictitious facility i_0 . As in the base model, new facilities cannot be downgraded.

The opposite scenario, i.e. capacity reduction, can also easily be incorporated in our model. In this case, a *fictitious facility*, denoted by j_0 , is considered which can be established with all excess capacity from existing facilities. This implies that j_0 has a similar status to that of the potential new facilities in \mathcal{S}^o , but cannot satisfy any demand requirements.

Another simple extension of our model is to consider modular capacity transfers instead of continuous shifts. Melo *et al.* [3] have shown that the base model along with the above extensions includes many well known dynamic facility location problems in which network design decisions are restricted to opening new facilities and closing existing ones. On the other hand, it generalizes many of the models proposed in the literature (see Melo *et al.* [3] for an extensive comparison). Currently, focus is being given to the development of heuristic procedures based on variable neighborhood search to decide on the facilities that should operate in each period. Preliminary results indicate that on average feasible solutions can be obtained within 2% of the optimum.

References

1. T. Bender, H. Hennes, J. Kalcsics, M. T. Melo, and S. Nickel. Location Software and Interface with GIS and Supply Chain Management. In Z. Drezner and H. W. Hamacher, editors, *Facility Location: Applications and Theory*, chapter 8, pages 233–274. Springer-Verlag, Berlin, Heidelberg, 2002.
2. J. Kalcsics, T. Melo, S. Nickel, and V. Schmid-Lutz. Facility Location Decisions in Supply Chain Management. In K. Inderfurth et al., editors, *Operations Research Proceedings 1999*, pages 467–472. Springer Verlag, Berlin, 2000.
3. M. T. Melo, S. Nickel, and F. Saldanha da Gama. Large-Scale Models for Dynamic Multi-Commodity Capacitated Facility Location. Technical report 58, Fraunhofer Institute for Industrial Mathematics (ITWM), Kaiserslautern, Germany, 2003. Available at www.itwm.fhg.de

Leistungsabstimmung von Produktionslinien in der Elektronikmontage

Schleusener, M., Günther, H.-O.

Fachgebiet Produktionsmanagement, Technische Universität Berlin, Wilmersdorfer Str. 148, D-10585 Berlin, Email: {m.schleusener, ho.guenther}@pm-berlin.net

Zusammenfassung Bei der Leistungsabstimmung von Produktionslinien in der Elektronikmontage (Leiterplattenbestückung) sind die zu bestückenden Bauelemente auf die verschiedenen Automaten innerhalb der Linie aufzuteilen. Hierbei sind sowohl die begrenzten Magazinkapazitäten als auch die technischer Restriktionen der Automaten zu beachten. Gleichzeitig sind unter Beachtung einer möglichen Mehrfachaufrüstung ausgewählter Bauelementtypen die Bestückungsoperationen auf die verschiedenen Automaten innerhalb der Linie aufzuteilen. Ziel ist der Belastungsausgleich zwischen den Automaten. Zur Lösung dieser komplexen Entscheidungsprobleme wird eine effiziente Heuristik vorgeschlagen.

1. Einleitung

Hochautomatisierten Montagelinien kommt bei der Großserien- und Massenproduktion von elektronischen Baugruppen (Leiterplattenbestückung) eine herausragende Bedeutung zu. Hierbei werden oft nur ein einziger Leiterplattentyp oder eine begrenzte Anzahl verwandter Leiterplattentypen in einer Linie montiert. Das zu bestückende Bauelementespektrum erfordert zumeist unterschiedliche Automatentypen. Nur so kann die geforderte Flexibilität und Produktivität erreicht werden. Darüber hinaus sind weitere Einrichtungen in die Montagelinie integriert. Von entscheidender Bedeutung für die Taktzeit und damit für die gesamte Produktivität der Montagelinie sind die zur Steuerung der Bestückungsprozesse der einzelnen Automaten sowie der Leistungsabstimmung der Linie eingesetzten Algorithmen.

Während bei der Kleinserienmontage, die durch eine hohe Vielfalt unterschiedlicher Leiterplattentypen geprägt ist, die Optimierung der Auftragsfreigabe und Auftragsreihenfolge im Vordergrund steht (vgl. [7], [8], [11]), sind bei Großserien- und Massenproduktion die Optimierung der Arbeitsweise der einzelnen Automaten und darüber hinaus die Leistungsabstimmung innerhalb einer Montagelinie von ausschlaggebender Bedeutung.

Im Wesentlichen enthält eine Montagelinie die folgenden Einrichtungen:

- verschiedene Arten von Bestückungsautomaten,
- Peripheriegeräte (z.B. Lotpastendrucker, Klebestation, Lötöfen, Reinigungsstation),

- Materialhandlinggeräte (z.B. Leiterplattenzufuhr, Wendestationen, Transportbänder, Puffer),
- Testeinrichtungen (z.B. Beschaffenheitsprüfung, Sichtprüfung, Funktionsprüfung) sowie u.U. eine Nacharbeitsstation.

Abb. 1 zeigt beispielhaft den Aufbau einer Montagelinie bei einem Hersteller von Telematiksystemen. Als erstes wird mit Hilfe eines Siebdruckverfahrens Lotpaste an den Bestückungspositionen aufgetragen. Die Lotpaste besteht aus einer Mischung von Lotzinn, Klebstoff und Flussmittel und dient dazu, die Bauelemente auf der Leiterplatte an ihren vorgesehenen Positionen zu fixieren. Durch verschiedene Bestückungsautomaten werden dann sog. oberflächenmontierte Bauelemente (SMDs, surface mount devices) bestückt. Anschließend werden mit Hilfe eines Lötovens durch Schmelzen des Lotzinns die Bauelemente mit der Leiterplatte elektrisch verbunden. Die Leiterplatten durchlaufen dann noch weitere Bestückungsautomaten, in denen Sonderformen von Bauelementen (u.a. Axial- und Radialkomponenten) montiert werden. Nach Wenden der Leiterplatte erfolgt ein weiterer Durchlauf durch die Montagelinie zur Bestückung von oberflächenmontierten Bauelementen auf der Rückseite der Leiterplatte.

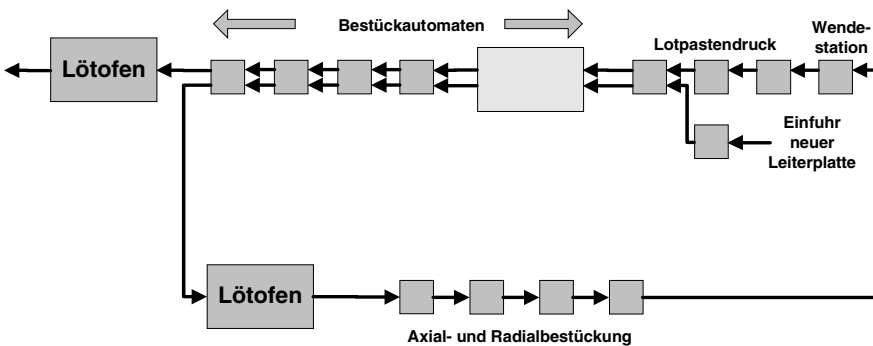


Abb. 1. Beispiel einer Montagelinie zur Bestückung von Leiterplatten

2. Problemstellung

Bei der Leistungsabstimmung von Montagelinien stehen zwei eng miteinander verknüpfte Probleme im Vordergrund, nämlich die Aufteilung der zu bestückenden Bauelemente auf die verschiedenen Automaten innerhalb der Linie unter Beachtung der begrenzten Magazinkapazitäten und der jeweiligen technischer Restriktionen der Automaten sowie die Zuordnung der Bestückungsoperationen auf die verschiedenen Automaten. Die Zielsetzung besteht darin, einen Belastungsausgleich zwischen den Automaten innerhalb der Linie zu erreichen. Die meisten industriellen Montagelinien sind so aufgebaut, dass die verschiedenen Hilfseinrichtungen (z.B. Lötoven) sich nicht als Engpass erweisen und daher bei der Leistungsabstimmung i.d.R. vernachlässigt werden können.

Im Gegensatz zu den aus der wissenschaftlichen Literatur bekannten klassischen Verfahren zur Leistungsabstimmung bei Fließproduktion ist in der Elektronikmontage die spezifische Arbeitsweise der eingesetzten Automaten zu beachten. Welche Erhöhung der Bestückzeit durch die Zuordnung eines Bestückungsvorgangs bzw. des zugehörigen Bauelements zu einem Automaten eintritt, lässt sich ohne eine exakte Analyse der kinematischen Abläufe und ohne Berücksichtigung der Anordnung der Bauelementezuführungen im Magazin des Automaten und der zur Automatensteuerung verwendeten Algorithmen nicht zuverlässig ermitteln (vgl. [4]).

So werden für die hochpräzise Bestückung komplexer Bauelemente sog. Pick & Place-Automaten eingesetzt, bei denen die Bauelemente jeweils einzeln durch einen frei beweglichen Roboterarm aus dem Magazin entnommen und anschließend auf der Leiterplatte aufgesetzt werden. Eine deutlich höhere Bestückungsleistung weisen Collect&Place-Automaten auf, bei denen der Roboterarm mit einem Revolver zur Aufnahme mehrerer Bauelemente ausgestattet ist, sowie Chip-Shooter, bei denen der Transfer der Bauelemente vom Magazin zur Leiterplatte durch ein rotierendes Karussell erfolgt. Schließlich wurden für die Bestückung von Standardbauelementen modulare Bestückungsautomaten entwickelt, bei denen verschiedene Pick&Place- oder Collect&Place-Module wie bei einer Transferstrasse sequentiell innerhalb des Automaten angeordnet sind. Diese Automaten erreichen die mit Abstand höchste Bestückungsleistung. Aus technischen Gründen können Bauelemente nur auf bestimmten Automaten bestückt werden. Daher ist eine Linie zumeist aus unterschiedlichen Automatentypen aufgebaut.

Welche Bestückungsleistung tatsächlich erreicht werden kann, hängt nicht nur von dem jeweiligen Automatentyp ab, sondern auch wesentlich von der gewählten Zuordnung der Bauelementtypen zu den Positionen im Magazin des Automaten sowie von der Reihenfolge der Bestückungsoperationen. Hierbei handelt es sich um komplexe Optimierungsprobleme, für die keine exakten Lösungsverfahren zur Verfügung stehen. In die Leistungsabstimmung von Montagelinien sind diese Gesichtspunkte der Automatensteuerung zu integrieren. Sowohl die aus der wissenschaftlichen Literatur bekannten Ansätze als auch die im kommerziellen Softwareangebot enthaltenen Algorithmen ignorieren diese Anforderung, indem sie von gegebenen Bestückzeiten und anderen vereinfachenden Annahmen ausgehen (vgl. z.B. [9], [10], [11], [12]). Andere Autoren beschränken sich auf die Analyse von Tandemkonfigurationen (z.B. [2]).

3. Mathematische Modellformulierung

Im Mittelpunkt der Leistungsabstimmung von Montagelinien steht der Belastungsausgleich zwischen den einzelnen Automaten. Hierzu wird im Folgenden ein mathematisches Optimierungsproblem formuliert, wobei von der vereinfachenden Annahme ausgegangen wird, dass die Bestückungszeit eines Bauelements auf den Automaten gegeben ist. Wie im Vorabschnitt erläutert, hängt diese Bestückungszeit von der Anordnung der Bauelementtypen im Magazin des Automaten und von

der Reihenfolge der Bestückungsoperationen ab. Die folgende Modellformulierung erfasst daher lediglich das Kernproblem der Leistungsabstimmung für einen gegebenen Leiterplattentyp. Eine effiziente Heuristik zur Berücksichtigung des Einflusses der Automatensteuerung wird im nächsten Abschnitt vorgestellt.

Indizes, Indexmengen

- $i \in I$ Bauelemente (Bestückungsoperationen)
 $j \in J$ Automaten in der Montagelinie
 $i \in I_j$ Bauelemente, die von Automat j bestückt werden können
 $t \in T_j$ Bauelementtypen, die von Automat j bestückt werden können

Parameter

- M_j Magazinkapazität von Automat j (maximale Anzahl an Bauelementtypen)
 $t(i)$ Bauelementtyp, der in Operation i bestückt wird
 c_{ij} Bestückungszeit für Bauelement i auf Automat j

Entscheidungsvariablen

- x_{ij} = 1, falls Bauelement i von Automat j bestückt wird (= 0, sonst)
 y_{jt} = 1, falls Bauelementtyp t auf Automat j gerüstet wird (= 0, sonst)
 w maximale Belastung eines Automaten in der Montagelinie

Modellformulierung

$$\min w \quad (1)$$

unter den Nebenbedingungen

Maximale Arbeitslast der Automaten

$$w \geq \sum_{i \in I_j} c_{ij} \cdot x_{ij} \quad j \in J \quad (2)$$

Magazinkapazität

$$\sum_{t \in T_j} y_{jt} \leq M_j \quad j \in J \quad (3)$$

Magazinrüstung

$$x_{ij} \leq y_{j,t(i)} \quad j \in J, i \in I_j \quad (4)$$

Bestückungsoperationen

$$\sum_{j \in J: i \in I_j} x_{ij} = 1 \quad i \in I \quad (5)$$

Wertebereiche

$$x_{ij} \in \{0,1\} \quad j \in J, i \in I_j \quad (6)$$

$$y_{jt} \in \{0,1\} \quad j \in J, t \in T_j \quad (7)$$

$$w \geq 0 \quad (8)$$

Zu minimieren ist die Taktzeit der Linie, die durch das Maximum der Arbeitslast aller Automaten der Linie bestimmt wird. Dies erfolgt durch die Zielfunktion (1) in Verbindung mit Nebenbedingung (2). Die beschränkte Kapazität der Bauelementemagazine der einzelnen Automaten wird durch (3) erfasst. Nebenbedingung (4) besagt, dass ein bestimmtes Bauelement von einem Automaten nur dann bestückt werden kann, wenn der zugehörige Bauelementtyp auf dem betreffenden Automaten gerüstet ist. (Bauelement i ist hierbei vom Typ $t(i)$). Nebenbedingung (5) stellt sicher, dass jedes Bauelement genau einmal bestückt wird. In (6) bis (8) werden die Wertebereiche der Entscheidungsvariablen definiert.

Das Modell (1) bis (8) stellt ein binäres Optimierungsmodell mit einer zusätzlichen kontinuierlichen Variablen dar, das für realistische Problemgrößen mehrere Tausend Variablen und Nebenbedingungen enthält. Die exakte Lösung des Modells wäre aber dennoch als Näherungslösung des realen Problems der Leistungsabstimmung anzusehen, da – wie oben erwähnt – die Bestückungszeiten als gegeben angenommen und nicht etwa durch eine explizite Optimierung der Automatensteuerung gewonnen werden.

4. Heuristisches Lösungsverfahren

Die Leistungsabstimmung der gesamten Montagelinie unter Berücksichtigung der Entscheidungen über die Zuordnung der Bauelementtypen zu den Magazinpositionen der Automaten und über die Reihenfolge der Bestückungsoperationen erweist sich als äußerst komplexes Optimierungsproblem, für das keine exakten Lösungsverfahren bekannt sind. Daher wird im Folgenden eine neuartige, effiziente Heuristik vorgeschlagen. Dieser Ansatz unterscheidet sich von den aus der Literatur bekannten Ansätzen durch die Einbindung der Automatenoptimierung mit einem sehr hohen Genauigkeitsgrad. Hierzu wurde basierend auf den Erfahrungen mit der Optimierung modularer Bestückungsautomaten (vgl. [6]) ein mehrstufiger Algorithmus entwickelt, der den Belastungsausgleich zwischen den Automaten der Linie und die Optimierung auf Automatenenebene integriert.

Im Einzelnen gehen wir von folgenden Annahmen aus:

- Die Arbeitsausführung innerhalb der Linie ist getaktet, wobei sich die Taktzeit aus der maximalen zeitlichen Belastung der einzelnen Automaten ergibt.
- Die Zuführeinrichtungen (Feeder) für die verschiedenen Bauelementtypen belegen jeweils nur einen Magazinplatz.
- Ein Bauelementtyp kann an mehreren Automaten gerüstet werden.

- Für jeden Bauelementtyp ist vorgegeben, auf welchen Automaten er bestückt werden kann.
- Innerhalb einer Linie gibt es unterschiedliche Automatentypen, die sich hinsichtlich des zulässigen Spektrums von Bauelementtypen, der Magazinkapazität, der Bestückungsgeschwindigkeit und insbesondere der kinematischen Eigenschaften unterscheiden.

Die Intention des heuristischen Lösungsverfahrens besteht darin, die Bestückungszeiten der Automaten nicht als gegeben anzunehmen, sondern diese Größen aus der Automatensteuerung unter Berücksichtigung der kinematischen Eigenschaften der Automaten abzuleiten. Das Verfahren durchläuft drei Stufen. Zur Erläuterung des Verfahrens wird beispielhaft eine Leiterplatte betrachtet, die mit 298 Bauelementen und insgesamt 43 Bauelementtypen zu bestücken ist. Die Montagelinie besteht aus je einem Pick&Place- und Collect&Place-Automaten sowie aus einem Chip-Shooter und einem modularen Bestückungsautomaten.

Stufe I: Bestimmung von Bestückungsfaktoren. Zunächst wird für den betrachteten Leiterplattentyp die effektive Bestückungsleistung der verschiedenen Automaten unter Berücksichtigung ihrer jeweiligen kinematischen Eigenschaften bestimmt. Hierzu wird hypothetisch davon ausgegangen, dass jeder Automat in der Montagelinie sämtliche Bestückungsoperationen ausführt. Somit werden die Restriktionen (3) bis (5), d.h. insbesondere auch die beschränkten Magazinkapazität der Automaten, zunächst vernachlässigt. Um die Bestückungszeiten auf den einzelnen Automaten realistisch zu ermitteln, werden die Magazinbelegung und die Bestückungssequenzen mit Hilfe spezifischer Verfahren zur Automatensteuerung bestimmt. Hierzu wird für Pick&Place- und Collect&Place-Automaten auf das Verfahren von Grunow et al. [5], für Chip-Shooter auf Grunow [3] sowie für modulare Bestückungsautomaten auf Grunow et al. [6] zurückgegriffen. Die erhaltenen Bestückungszeiten pro Bauelement lassen sich relativ zu denjenigen des schnellsten Automaten als Bestückungsfaktoren angeben. Tabelle 1 zeigt die Ergebnisse der ersten Stufe für die ausgewählte Leiterplatte.

Tab. 1. Ergebnisse der ersten Stufe der Heuristik

Automat	Bestückungszeit (Sekunden)	Anzahl Bauelementtypen	Anzahl Bestückungsoperationen	Bestückungsfaktor
Modular	22,86	43	298	1,00
Collect&Place	79,06	43	298	3,46
Pick&Place	362,76	43	298	15,87
Chip-Shooter	28,28	43	298	1,24

Stufe II: Berücksichtigung der Magazinplatzkapazität der Automaten. In der zweiten Stufe des Verfahrens wird eine Lösung ermittelt, bei der die begrenzten Magazinkapazitäten der Automaten berücksichtigt und nur solche Bestückungsoperationen zugelassen werden, bei denen die zugehörigen Bauelementtypen auch auf dem Automaten gerüstet sind (Berücksichtigung der Nebenbedingungen (3) und (4) des Optimierungsmodells). Hierbei werden auch die Einschränkungen bezüglich des für einen Automaten technisch zulässigen Bauelementespektrums beachtet.

Im Einzelnen werden die folgenden Schritte durchlaufen:

1. Für jede technisch zulässige Bauelementtyp-Automaten-Kombination wird ein Prioritätswert als Quotient aus der Anzahl der von dem Bauelementtyp zu bestückenden Bauelemente und dem Bestückungsfaktor des betreffenden Automaten bestimmt.
2. Nach Maßgabe des maximalen Prioritätswerts aller technisch zulässigen Bauelementtyp-Automaten-Kombination und unter Beachtung der begrenzten Magazinkapazitäten werden die Bauelementtypen und somit auch die zugehörigen Bestückungsoperationen den jeweiligen Automaten zugeordnet. (Man beachte, dass in diesem Schritt Bestückungsoperationen vorerst noch mehreren Automaten zugeordnet sind, Nebenbedingung (5) des Optimierungsmodells also noch vernachlässigt wird.)
3. Die Automatensteuerung wird aktualisiert, d.h. aufgrund der den Automaten zugeordneten Bauelementtypen und Bestückungsoperationen werden eine neue Magazinbelegung und neue Bestückungssequenzen ermittelt. Hierzu kommen wie zuvor die Verfahren von Grunow et al. [5], Grunow [3] sowie Grunow et al. [6] zum Einsatz. Gleichzeitig werden neue Bestückungsfaktors bestimmt.

Die Ergebnisse der zweiten Stufe des Verfahrens sind in Tabelle 2 dargestellt.

Tab. 2. Ergebnisse der zweiten Stufe der Heuristik

Automat	Bestückungszeit (Sekunden)	Anzahl Bauelementtypen	Anzahl Bestückungsoperationen	Bestückungsfaktor
Modular	22,86	43	298	1,00
Collect&Place	27,61	5	97	3,71
Pick&Place	34,57	1	25	18,03
Chip-Shooter	22,37	22	242	1,26

Stufe III: Belastungsausgleich zwischen den Automaten. Abschließend wird jede Bestückungsoperation definitiv nur einem einzigen Automaten zugeordnet (Berücksichtigung von Nebenbedingung (5) des Optimierungsmodells). Hierzu wird ein einfaches binäres Optimierungsmodell aufgestellt und mit Hilfe von Standardsoftware innerhalb weniger Sekunden gelöst. In der Magazinrüstung eines Automaten verbleiben nur diejenigen Bauelementtypen, die auch tatsächlich auf dem Automaten bestückt werden. Danach werden für jeden Automaten erneut die Magazinbelegung und die Bestückungssequenzen aktualisiert. Man erhält die in Tabelle 3 dargestellte Abschlusslösung. Die Taktzeit der Linie ergibt sich aus der Bestückungszeit des Chip-Shooters von 13,89 Sekunden. Der am wenigsten belastete Automat weist immerhin noch eine Auslastung von 96,11% auf.

Tab. 2. Ergebnisse der dritten Stufe der Heuristik

Automat	Bestückungszeit (Sekunden)	Anzahl Bauelementtypen	Anzahl Bestückungsoperationen	Bestückungsfaktor
Modular	13,68	31	101	1,00
Collect&Place	13,35	4	46	3,71
Pick&Place	13,80	1	10	18,03
Chip-Shooter	13,89	18	141	1,26

4. Fazit

Die Leistungsabstimmung von Montagelinien zur Leiterplattenbestückung stellt in der industriellen Praxis ein wichtiges, aber häufig unbefriedigend gelöstes Planungsproblem dar. Die in kommerziell angebotener Software implementierten Verfahren gehen ebenso wie die bisher in der Literatur veröffentlichten Ansätze von einer gegebenen Bestückungsleistung der Automaten aus. Tatsächlich hängt die effektive Bestückungsleistung der Automaten ganz wesentlich von der Breite des Bauelementespektrums und der Anzahl von Bauelementen je Leiterplatte sowie der zur Automatensteuerung eingesetzten Algorithmen ab. Daher wurde ein Verfahren entwickelt, das diese Einflussfaktoren integrativ erfasst. Weitere numerische Untersuchungen zur Analyse der Güte des Verfahrens stehen noch aus.

Literatur

1. Gronalt M, Grunow M, Günther HO, Zeller R (1997) A heuristic for component switching on SMT placement machines. *International Journal of Production Economics*, 53: 181-190
2. Gronalt M, Zeller R (1998) Job sequencing and component setup on a surface mount placement machine. *Production Planning and Control* 9: 201-211
3. Grunow M (2000) Optimierung von Bestückungsprozessen in der Elektronikmontage. Gabler – Deutscher Universitätsverlag, Wiesbaden
4. Grunow M, Günther HO, Föhrenbach A (2000) Simulation-based performance analysis and optimization of electronics assembly equipment. *International Journal of Production Research* 38: 4247-4259
5. Grunow M, Günther HO, Schleusener M (2003a) Component allocation for printed circuit board assembly using modular placement machines. *International Journal of Production Research*, 41: 1311-1331
6. Grunow M, Günther HO, Schleusener M, Yilmaz IO (2003b) Optimizing operations of a collect-and-place machine in PCB assembly. Working paper, TU Berlin
7. Günther HO, Gronalt M, Zeller R (1998) Job sequencing and component setup on a surface mount placement machine. *Production Planning and Control* 9: 201-211
8. Günther HO, Grunow M, Schorling Ch (1997) Workload planning in small lot printed circuit board assembly. *OR Spektrum* 19: 147-157
9. Hillier M, Brandeau (2001) Cost minimization and workload balancing in printed circuit board assembly. *IIE Transactions* 33: 547-557
10. Ji P, Sze MT, Lee WB (2001) A genetic algorithm of determining cycle time for printed circuit board assembly lines. *European Journal of Operational Research* 128: 175-184
11. Lapierre S, Debargis L, Soumis F (2000) Balancing printed circuit board assembly line systems. *International Journal of Production Research* 38: 3899-3911
12. Sawik T (2002) Balancing and scheduling of surface mount technology lines. *International Journal of Production Research* 40: 1973-1991

A Priority-Rule Based Method for Batch Production Scheduling in the Process Industries

Christoph Schwindt¹ and Norbert Trautmann²

¹ Universität Halle-Wittenberg,
Institut für Wirtschaftsinformatik und Operations Research,
D-06099 Halle (Saale), Germany,
e-mail: schwindt@wior.uni-karlsruhe.de

² Universität Karlsruhe, Institut für Wirtschaftstheorie und Operations Research,
D-76128 Karlsruhe, Germany,
e-mail: trautmann@wior.uni-karlsruhe.de

Abstract. We present a priority-rule based method for batch production scheduling in the process industries. The problem consists in scheduling a given set of operations on a batch plant such that the makespan is minimized. Each operation consumes given amounts of input materials at its start and produces given amounts of output materials at its completion. An initial, a maximum, and a minimum stock level are specified for each material. Some materials are perishable and thus cannot be stored. The operations may be executed on alternative processing units. Processing units require cleaning between certain operations and before any idle time. Due to the constraints on material availability and storage capacity, classical schedule-generation schemes cannot be applied to this problem. That is why we propose a new two-phase approach dealing with the two types of constraints separately.

1 Introduction

In the process industries, the production of final products takes place by running various processes (hereinafter referred to as *tasks*) that change the attributes or composition of the input. In some of the most important branches of the process industries, e.g. in the life science industries, customers typically order low volumes of a great variety of products. In those industries, production is typically carried out on multi-purpose plants operated in batch production mode. A multi-purpose plant consists of *processing units* that can operate different tasks. Some tasks can be executed on alternative processing units, which may differ in processing and cleaning times. In general, raw materials, intermediates, and final products can be stocked in *storage facilities*. In batch production mode, all material flows are discontinuous, where for simplicity we suppose that the input of a task is consumed at the start, and the output becomes available at the completion of the task. A *batch* associates a task with the respective input and output quantities consumed and produced. The processing of a batch is called an *operation*. A task may be

carried out more than once, resulting in several operations. The processing time of an operation is assumed to be independent of the batch size.

In what follows, we illustrate some features typical of batch production in the process industries using a case study for short-term production scheduling presented in Westenberger and Kallrath [8]. Given the primary requirements for all final products, *short-term production scheduling* is concerned with finding a set of batches and a detailed occupancy plan for the multi-purpose plant in such a way that the production makespan is minimized. The following extended version of the state-task network (STN) concept (cf. Kondili et al. [4]) allows for a concise description of short-term production scheduling problems. An STN is a directed graph containing three types of elements:

- *State nodes* represent the raw materials, intermediates, and final products. They are drawn as ovals indicating the state number, and the initial, minimum, and maximum stock levels of the corresponding product. Some of the intermediate products cannot be stocked. In this case the oval is labelled with “ns”. The value ∞ for the initial or maximum stock levels means that there is sufficient initial stock or unlimited storage capacity available, respectively.
- *Task nodes* stand for the processes, which transform one or more input products into one or more output products. The task nodes are represented as rectangles labelled with the task number, the interval of feasible batch sizes β , the required processing unit U , as well as the processing/cleaning times of the task. If a task can be executed on alternative processing units, the corresponding processing and cleaning times are listed for each unit. In the case study, a processing unit needs to be cleaned after the completion of a task if one passes to a task with a higher number or if no task is started immediately after the completion of the preceding one.
- *Arcs* linking state and task nodes belong to material flows. If more than one input product is consumed or more than one output product is produced, the corresponding input or output proportions are specified on the arcs. For some tasks, the input or output proportions can be chosen by varying certain process parameters. In this case, the arcs are labelled with the intervals of feasible proportions x .

Figure 1 shows the STN for the batch production of the case study with 19 products, 17 tasks, and 9 processing units. Alternative processing units are available for processing tasks 10 to 14, 16, and 17.

Several authors have presented solution methods for short-term production scheduling in process industries and have used the above case study to compare the performance of their planning methods. Most of the approaches are based on the time-indexed formulation of the planning problem as a mixed-integer linear program presented by Kondili et al. [4]. Blömer and Günther [1], [2] develop various heuristic methods for reducing the number of decision variables. Burkard et al. [3] apply rounding heuristics for the case

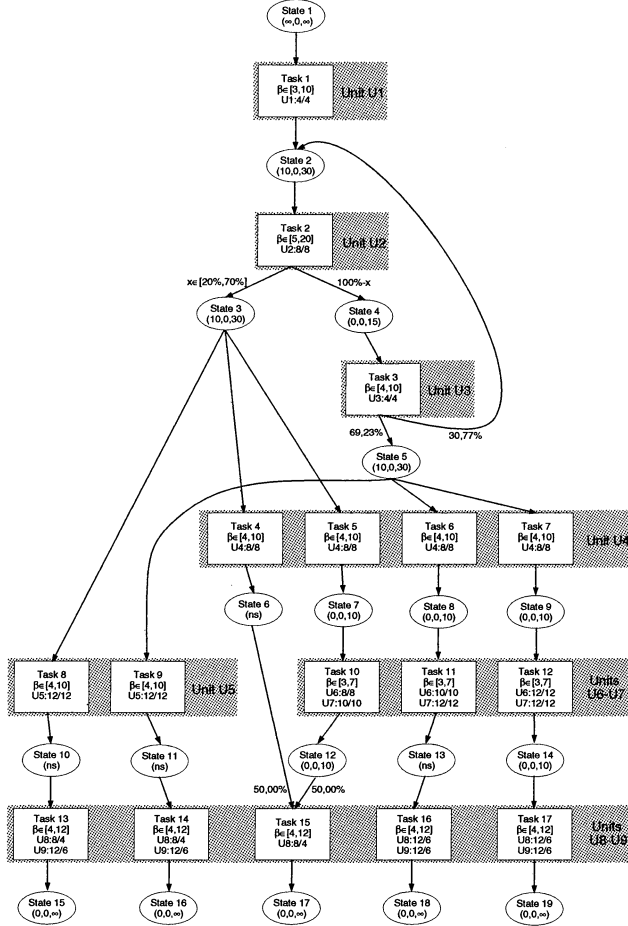


Fig. 1. State-task network for case study

without cleaning times. Neumann et al. [6] decompose the short-term production scheduling problem into a batching and a batch (production) scheduling problem. A solution to the *batching problem* provides the numbers and sizes of batches for all tasks subject to inventory balance and storage capacity constraints. This problem can be formulated as a mixed-binary linear program of moderate size (cf. Neumann et al. [6]). For a given set of operations, the *batch scheduling problem* is concerned with the scheduling of the operations on the processing units subject to inventory balance and storage capacity constraints. Neumann et al. [6] develop a truncated branch-and-bound method for solving the batch scheduling problem. Following the lines of the decomposition approach, in this paper we present a priority-rule based method for the latter problem.

2 Problem Statement

In this section, we outline a formal statement of the batch scheduling problem. For details we refer to Neumann et al. [6].

Suppose that n operations $1, \dots, n$ have to be scheduled. For notational convenience we introduce two fictitious operations 0 and $n+1$ representing the production start and the production end, respectively. $\tilde{V} := \{1, \dots, n\}$ is the set of all real operations, and $V := \tilde{V} \cup \{0, n+1\}$ is the set of all operations. Let $S_i \geq 0$ be the sought *start time* of operation i . S_{n+1} coincides with the production makespan. Vector $S = (S_i)_{i \in V}$ with $S_0 = 0$ is called a *schedule*.

Each **processing unit** can be viewed as a *renewable resource* (cf. Neumann et al. [7], Sect. 2.1) of capacity 1. Let \mathcal{R}^ρ be the set of all renewable resources and \mathcal{R}_i^ρ be the set of those alternative renewable resources on which operation i can be carried out. For $i \in \tilde{V}$ and $k \in \mathcal{R}_i^\rho$, the binary decision variable x_{ik} indicates whether or not resource k processes operation i ($x_{ik} = 1$ or $x_{ik} = 0$, respectively). Each real operation i must be executed on exactly one processing unit, i.e., $\sum_{k \in \mathcal{R}_i^\rho} x_{ik} = 1$ ($i \in \tilde{V}$). Vector $x = (x_{ik})_{i \in \tilde{V}, k \in \mathcal{R}_i^\rho}$ is then called an *assignment* of operations i to processing units k . By $p_i(x)$ and $c_i(x)$ we denote the processing and cleaning times of activity i given assignment x , where we suppose that $p_i(x) = c_i(x) = 0$ for $i = 0, n+1$.

The need for cleaning a processing unit generally depends on the operations sequence on this unit. Let $P_k \subseteq \tilde{V} \times V$ be the set of operation pairs (i, j) for which passing from i to j requires a cleaning of processing unit k . Given a schedule S and an assignment x , let $O_k(S, x)$ designate the set of all pairs (i, j) such that $i \neq j$, $S_i \leq S_j$, and $x_{ik} = x_{jk} = 1$. $O_k(S, x)$ can be partitioned into the set $C_k(S, x)$ containing all pairs (i, j) for which k has to be cleaned between i and j (because $(i, j) \in P_k$ or $S_j > S_i + p_i(x)$) and the set $\overline{C}_k(S, x)$ of pairs for which j must be started immediately after the completion of operation i . A schedule S is called *process-feasible* with respect to assignment x if

$$\left. \begin{aligned} S_j &\geq S_i + p_i(x) + c_i(x), & \text{if } (i, j) \in C_k(S, x) \\ S_j &= S_i + p_i(x), & \text{if } (i, j) \in \overline{C}_k(S, x) \end{aligned} \right\} \quad (k \in \mathcal{R}^\rho) \quad (1)$$

Now we turn to the **storage facilities**, which represent so-called *cumulative resources* (cf. Neumann and Schwindt [5]). For each nonperishable product, there is one cumulative resource keeping its inventory. Let \mathcal{R}^γ be the set of all cumulative resources. For each $k \in \mathcal{R}^\gamma$, a minimum inventory \underline{R}_k (safety stock) and a maximum inventory \overline{R}_k (storage capacity) are given. Each operation $i \in V$ has a demand r_{ik} for resource $k \in \mathcal{R}^\gamma$. If $r_{ik} > 0$, the inventory of resource k is replenished by r_{ik} units at time $S_i + p_i(x)$. If $r_{ik} < 0$, the inventory is depleted by $-r_{ik}$ units at time S_i . r_{0k} represents the initial stock level of resource k . Let $V_k^+ := \{i \in V \mid r_{ik} > 0\}$ and $V_k^- := \{i \in V \mid r_{ik} < 0\}$ be the sets of operations replenishing and depleting,

respectively, the inventory of $k \in \mathcal{R}^\gamma$. Schedule S is said to be *storage-feasible* with respect to assignment x if

$$\underline{R}_k \leq \sum_{i \in V_k^+ : S_i + p_i(x) \leq t} r_{ik} + \sum_{i \in V_k^- : S_i \leq t} r_{ik} \leq \bar{R}_k \quad (k \in \mathcal{R}^\gamma, t \geq 0) \quad (2)$$

Usually **temporal constraints** of the type $S_j \geq S_i + \delta_{ij}(x)$ for $(i, j) \in E$ with $E \subseteq V \times V$ have to be taken into account as well. $\delta_{ij}(x)$ is a *minimum time lag* between the start of activities i and j . If $\delta_{ij}(x) < 0$, then $-\delta_{ij}(x)$ can be interpreted as a *maximum time lag* between the start of operations j and i . In case of $\delta_{ij}(x) = p_i(x)$, the corresponding temporal constraint is referred to as a *precedence constraint*. For each operation $i \in V$ we set $\delta_{0i}(x) := 0$ and $\delta_{i,n+1}(x) := p_i(x) + c_i(x)$. Based on time lags $\delta_{ij}(x)$ for $(i, j) \in E$ we can compute *distances* $d_{ij}(x)$ between any two activities $i, j \in V$. Distances $d_{ij}(x)$ coincide with the minimum time lags between activities i and j that are implied by the prescribed time lags (see e.g., Neumann et al. [7], Sect. 1.3). Given an assignment x , a schedule S satisfying $S_j \geq S_i + \delta_{ij}(x)$ for all $(i, j) \in E$ is called *time-feasible* with respect to x .

A schedule which is time-, process-, and storage-feasible with respect to some assignment x is called *feasible* with respect to x . A pair (S, x) is a feasible solution to the batch scheduling problem if x is an assignment and S is a feasible schedule with respect to x . The batch scheduling problem consists in finding a feasible solution (S, x) with minimum makespan S_{n+1} .

3 Solution Method

The basic idea of the new priority-rule based solution method is as follows. At first, we choose an assignment x of operations to processing units, where we balance the workload to be processed on alternative processing units by using a simple greedy heuristic. The method then consists of two phases. During the first phase, we relax the storage-capacity constraints. Using a serial schedule-generation scheme, the operations are iteratively scheduled on the processing units in such a way that the inventory does not fall below the safety stock at any point in time. Deadlocks are avoided by means of a specific unscheduling technique. Based on the resulting schedule, precedence constraints between replenishing and depleting operations are introduced according to a FIFO strategy. Those precedence constraints ensure that the material-availability constraints are always satisfied. In the second phase, which again applies the serial schedule-generation scheme, the operations are scheduled subject to the storage-capacity and the precedence constraints introduced.

In the balance of this section we explain the schedule-generation scheme of the first phase in more detail. We then briefly sketch the modifications needed for using the scheme in the second phase. Since assignment x has been fixed, we from now on omit x in the notation of processing times, time lags, and inventory levels.

Let $Pred(j)$ be the set of predecessors of node j with respect to the strict order $\{(i, j) \in V \times V \mid d_{ij} \geq 0 \text{ and } d_{ji} < 0\}$. $i \in Pred(j)$ means that activity i must be started no later than activity j but that conversely, activity j may be started after activity i . Moreover, denote the *completed set* of activities i already scheduled by \mathcal{C} and let $S^{\mathcal{C}} := (S_i)_{i \in \mathcal{C}}$ be the *partial schedule* constructed. We say that an activity $j \notin \mathcal{C}$ is *eligible* for being scheduled if (i) all of its predecessors have been scheduled, i.e., $Pred(j) \subseteq \mathcal{C}$ and (ii), the inventory levels in all cumulative resources do not fall below the safety stocks after the completion of all activities from set $\mathcal{C} \cup \{j\}$, i.e., $r_k(S^{\mathcal{C}}, \infty) + r_{jk} \geq \underline{R}_k$ for all $k \in \mathcal{R}^\gamma$.

The procedure is now as follows (see Algorithm 1). At first, we initialize the earliest and latest start times ES_i and LS_i for all $i \in V$. In each iteration of the schedule-generation scheme we then determine the set \mathcal{E} of eligible activities j , select one activity $j^* \in \mathcal{E}$ according to priority indices $\pi(j)$, determine the earliest feasible start time $t^* \geq ES_{j^*}$ for activity j^* , schedule j^* at time t^* , and update the earliest and latest start times of the activities i not yet scheduled. Starting with partial schedule $S^{\mathcal{C}}$ where $\mathcal{C} = \{0\}$ and $S_0 = 0$ we perform those steps until all activities have been scheduled, i.e., until $\mathcal{C} = V$.

Sometimes it may happen that due to maximum time lags between scheduled activities $i \in \mathcal{C}$ and the activity j^* selected, the latest start time LS_{j^*} of j^* is strictly smaller than time t^* . Then no feasible start time can be found for activity j^* , and $S^{\mathcal{C}}$ cannot be extended to a feasible schedule. In this case, we perform the following unscheduling step. At first, we determine the set $\mathcal{U} = \{i \in \mathcal{C} \mid LS_{j^*}^* = S_i - d_{j^*i}\}$ of all activities i that must be delayed in order to increase the latest start time of j^* . Then, we increase the earliest start times of activities i from set \mathcal{U} to time $S_i + t^* - LS_{j^*}$, update the distances d_{ij} accordingly, and restart the scheduling procedure. In the implementation shown in Algorithm 1, the number u of unscheduling steps is limited by some upper bound \bar{u} .

After having obtained a time- and process-feasible schedule satisfying the material-availability constraints, we link producing and consuming operations following a FIFO strategy. This means that for each $k \in \mathcal{R}^\gamma$ we iterate the replenishing activities $i \in V_k^+$ according to nondecreasing completion times $S_i + p_i$ and allot the corresponding r_{ik} units to depleting activities $j \in V_k^-$ in the order of nondecreasing start times S_j . For each pair $(i, j) \in V_k^+ \times V_k^-$ for which j consumes units produced by i , we introduce a precedence constraint between i and j by setting $\delta_{ij} := \max(d_{ij}, p_i)$. Subsequently, we update the distances d_{ij} and proceed with the second phase of our procedure.

When during the second phase we deal with storage-capacity instead of material-availability constraints, we define the eligible set to be $\mathcal{E} := \{j \in V \setminus \mathcal{C} \mid Pred(j) \subseteq \mathcal{C}, r_k(S^{\mathcal{C}}, \infty) + r_{jk} \leq \bar{R}_k \text{ for all } k \in \mathcal{R}^\gamma\}$. In the definition of \mathcal{E} , we use the predecessor sets $Pred(j)$ from the first phase in order to allow for the scheduling of depleting operations before the replenishing operations

Algorithm 1 Schedule-generation scheme: Ensuring material availability

```

 $u := 0;$ 
2:  $S_0 := 0, \mathcal{C} := \{0\};$ 
   for all  $i \in V$  do (* initialize  $ES_i$  and  $LS_i$  *)
      $ES_i := d_{0i}, LS_i := -d_{i0};$ 
   while  $\mathcal{C} \neq V$  do
      $\mathcal{E} := \{j \in V \setminus \mathcal{C} \mid Pred(j) \subseteq \mathcal{C}, r_k(S^{\mathcal{C}}, \infty) + r_{jk} \geq \underline{R}_k \text{ for all } k \in \mathcal{R}^\gamma\};$ 
      $j^* := \min\{j \in \mathcal{E} \mid \pi(j) = \text{ext}_{h \in \mathcal{E}} \pi(h)\};$ 
      $t' := \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{j^*k} \geq \underline{R}_k \text{ for all } k \in \mathcal{R}^\gamma, \tau \geq t\};$ 
      $t^* := \min\{S_{j^*} \geq t' \mid S^{\mathcal{C} \cup \{j^*\}} \text{ is process-feasible}\};$ 
     if  $t^* > LS_{j^*}$  then (* unschedule and restart *)
        $u := u + 1;$ 
       if  $u > \bar{u}$  then terminate;
        $\mathcal{U} := \{i \in \mathcal{C} \mid LS_{j^*} = S_i - d_{j^*i}\};$ 
       for all  $i \in \mathcal{U}$  do  $d_{0i} := S_i + t^* - LS_{j^*};$ 
       update distances  $d_{ij}$  for all  $i, j \in V$  and goto line 2;
     else (* schedule  $j^*$  at time  $t^*$  *)
        $S_{j^*} := t^*, \mathcal{C} := \mathcal{C} \cup \{j^*\};$ 
       for all  $j \in V \setminus \mathcal{C}$  do (* update  $ES_j$  and  $LS_j$  *)
          $ES_j := \max(ES_j, S_{j^*} + d_{jj^*});$ 
          $LS_j := \min(LS_j, S_{j^*} - d_{jj^*});$ 
   return  $S;$ 

```

allotted to them have been added to the partial schedule. The earliest storage-feasible start time of activity j^* is now given by $t' := \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{jk} \leq \bar{R}_k \text{ for all } k \in \mathcal{R}^\gamma, \tau \geq t + p_{j^*}\}$. In this way, we ensure that any partial schedule $S^{\mathcal{C}}$ is feasible.

4 Experimental Performance Analysis

We have compared a randomized multi-pass version of the new priority-rule based method (PRM) to two alternative solution procedures from literature: the time grid heuristic (TGH) by Blömer and Günther [1] and the truncated branch-and-bound procedure (TBB) for batch scheduling devised in Neumann et al. [6]. We have used the test set introduced by Blömer and Günther, which has been constructed by varying the primary requirements for final products in the case study described in Section 1. For each instance, we have computed an optimal solution to the batching problem. This solution defines the batch scheduling problem that is (approximately) solved by the PRM and TBB heuristics. The results obtained for the 22 problem instances are displayed in Table 1, where a star marks those instances for which the priority-rule based method has been able to improve upon the best makespan known thus far. The results indicate that in particular for large problem instances, the new method compares favorably with the best algorithms from literature.

Table 1. Computational results

In- stance	Primary requirements	TGH S_{n+1} ^a	TGH time ^b	TBB S_{n+1} ^a	TBB time ^c	PRM S_{n+1} ^a	PRM time ^c
1	(20,20,20,0,0)	36	1110	36	3	39	60
2	(20,20,0,20,0)	42	2247	38	13	47	60
3	(20,20,0,0,20)	42	2487	38	17	48	60
4	(20,0,20,20,0)	48	1550	39	60	42	60
5	(20,0,20,0,20)	44	1778	41	60	41	60
6	(20,0,0,20,20)	48	3605	43	60	49	60
7	(0,20,20,20,0)	52	2587	38	60	44	60
8	(0,20,20,0,20)	48	3123	39	60	43	60
9	(0,20,0,20,20)	54	3607	53	60	49 *	60
10	(0,0,20,20,20)	60	3607	50	60	54	60
11	(10,10,20,20,30)	68	3605	66	60	64 *	60
12	(30,20,20,10,10)	60	3605	52	60	48 *	60
13	(10,20,30,20,10)	64	3604	50	60	57	60
14	(18,18,18,18,18)	66	3606	57	60	61	60
15	(15,15,30,30,45)	148	3622	114	60	100 *	60
16	(45,30,30,15,15)	124	3628	80	60	80	60
17	(15,30,45,30,15)	112	3621	91	60	92	60
18	(27,27,27,27,27)	124	3631	91	60	90	60
19	(20,20,40,40,60)	208	5152	135	60	159	60
20	(60,40,40,20,20)	184	3638	100	60	97 *	60
21	(20,40,60,40,20)	184	3643	112	60	124	60
22	(36,36,36,36,36)	214	3635	134	60	114 *	60

^a best makespan found^b CPU time in seconds on a Pentium-266 PC, procedure stopped after one hour if feasible solution could be found^c CPU time in seconds on a Pentium-800 PC, procedure stopped after one minute

References

1. Blömer, F., Günther, H.-O. (1998): Scheduling of a multi-product batch process in the chemical industry. *Computers in Industry* 36, 245–259
2. Blömer, F., Günther, H.-O. (2000): LP-based heuristics for scheduling chemical batch processes. *International Journal of Production Research* 35, 1029–1052
3. Burkard, R.E., Kocher, M., Rudolf, R. (1998): Rounding strategies for mixed integer programs arising from chemical production planning. *Yugoslav Journal of Operations Research* 8, 9–23
4. Kondili, E., Pantelides, C.C., Sargent, R.W.H. (1993): A general algorithm for short-term scheduling of batch operations: I. MILP Formulation. *Computers and Chemical Engineering* 17, 211–227
5. Neumann, K., Schwindt, C. (2002): Project scheduling with inventory constraints. *Mathematical Methods of Operations Research* 56, 513–533
6. Neumann, K., Schwindt, C., Trautmann, N. (2002): Advanced production scheduling for batch plants in process industries. *OR Spectrum* 24, 251–279
7. Neumann, K., Schwindt, C., Zimmermann, J. (2003): *Project Scheduling with Time Windows and Scarce Resources*, 2nd ed. Springer, Berlin
8. Westenberger, H., Kallrath, J. (1995): Formulation of a job shop problem in process industry. Bayer AG, Leverkusen and BASF AG, Ludwigshafen

A Metaheuristic Approach for Hazardous Materials Transportation

Pasquale Carotenuto¹, Graziano Galiano², and Stefano Giordani²

¹ C.N.R. - Istituto di Tecnologie Industriali e Automazione
via del Fosso del Cavaliere, 100 - 00133 Rome, Italy
E-mail: p.carotenuto@itia.cnr.it

² University of Rome "Tor Vergata" - Dipartimento di Informatica, Sistemi e Produzione
via del Politecnico, 1 - 00133 Rome, Italy
E-mail: giano76@tiscali.it; giordani@disp.uniroma2.it

Abstract. The transportation of hazardous materials is a growing problem due to the increasing transported volumes. What differentiates hazardous material shipments from shipments of other materials is the risk associated with an accidental release of hazardous materials during transportation. A possible solution to reduce the occurrence of dangerous events is to provide travel plans that establish a fair spatial and temporal distribution of the risk. The objective of this work is to study the problem of routing and scheduling a set of hazardous materials shipments, minimizing the travel total risk while spreading the risk among different zones of the geographical region where the transportation network is defined. We propose a genetic algorithm that, given a set of dissimilar routes for every origin-destination pair, selects a route and defines a departure time for every shipment with the aim of minimizing the total risk of the travel plans. The genetic algorithm is experimentally evaluated on a set of realistic scenarios defined on a regional area.

1 Introduction

In recent years there has been increased public and governmental concern regarding hazardous materials management as the production of such materials has increased. Hazardous materials, or dangerous goods, include explosives, gases, flammable liquids and solids, oxidizing substances, poisonous and infectious substances, radioactive materials, corrosive substances, and hazardous wastes: all important substances for industrial societies. Unfortunately, most hazardous materials are not used at their point of production, and they are transported over considerable distances. What differentiates shipments of hazardous materials from shipments of other materials is the risk associated with an accidental release of these materials during transportation. This can be extremely dangerous both with respect to the environment and to human health, since exposure to their toxic chemical ingredients could lead to the injury of plants, animals and humans. For these reasons, a carefully selection of a set of routes that involves less total risk and an equity distribution of the risk over the geographical region in which the transportation

network is embedded is necessary to prevent accidents, or, if they happen, to minimize their impact.

The modeling of transportation problems is a popular application area in OR. In most transportation planning models, the objective is to move products from origins to destinations at minimal cost. However, for hazardous materials shipments, a cost minimizing objective is usually not suitable. The risk associated with hazardous materials makes these problems more complicated than many other transportation problems. There are several excellent review articles which address the literature related to modeling of risk for hazardous materials transportation; however, there is no universally definition of risk. In this paper we refer to the traditional definition of risk over a link, that is the societal risk defined as the product of the population along the link within the neighborhood and the probability of an incident [2].

Several models have been proposed in literature that allow to determine routes of minimum risk under the tie of equity. The concept of dissimilar paths has also been considered in order to guarantee the spreading of risk [1]. Indeed, these methods address only the hazardous materials routing problem, allowing only to spatially spread equitably the risk over a region, but they do not consider the spreading the risk equitably over time.

This paper deals with the problem of routing and scheduling a set of hazardous materials shipments, minimizing the travel total risk while spreading the risk among different zones of the geographical region where the transportation network is defined, and also over time. We propose a genetic algorithm that, given a set of dissimilar routes for every origin-destination pair, selects a route and defines a departure time for every shipment with the aim of minimizing the total risk of the travel plans. The genetic algorithm is experimentally evaluated on a set of realistic scenarios defined on a regional area.

2 Problem Definition

We model a transportation network by a graph $G = (N, L)$, where N is the set of nodes representing relevant locations in the geographical area, and L is the set of links. Each link $l \in L$ has two weights associated with it, that is, the travel time t_l , and the risk index ρ_l , respectively.

Let S be a set of shipments. For each shipment $s \in S$, let $o_s \in N$, $d_s \in N$, pdt_s , be the origin node, the destination node and the preferred departure time, respectively. Let P_s be the given set of dissimilar routes (with each path p defined by a subset L_p of links in L) connecting node o_s to node d_s and let $r(p) = \sum_{l \in L_p} \rho_l$ be the risk of the route $p \in P_s$.

Let x_{sp} be a decision variable equal to 1 if the route $p \in P_s$ is assigned to the shipment s , and 0 otherwise. The objective is, for each shipment $s \in S$, to select a route $p \in P_s$ and assign a departure time $dt_s \in [pdt_s - \Delta, pdt_s + \Delta]$ that solve the following optimization model:

$$\min z = \sum_{s \in S} \sum_{p \in P_s} r(p) x_{sp} + \alpha \sum_{l \in L} \rho_l c_l(\mathbf{dt}, \mathbf{x}) \quad (1)$$

subject to

$$\sum_{p \in P_s} x_{sp} = 1, \quad \forall s \in S \quad (2a)$$

$$dt_s \geq p dt_s - \Delta, \quad \forall s \in S \quad (2b)$$

$$dt_s \leq p dt_s + \Delta, \quad \forall s \in S \quad (2c)$$

$$dt_s \geq 0, \quad \forall s \in S \quad (2d)$$

$$x_{sp} \in \{0, 1\}, \quad \forall s \in S, \forall p \in P. \quad (2e)$$

In (1), $\sum_{s \in S} \sum_{p \in P_s} r(p) x_{sp}$ represents the total risk of the plan, while $\sum_{l \in L} \rho_l c_l(\mathbf{dt}, \mathbf{x})$ denotes the risk associated to *conflicts* arising from simultaneous traversing of a link (or close links) as a function of the travel plan.

In the latter expression, $c_l(\mathbf{dt}, \mathbf{x})$ is the number of conflicts on link l . Given the shipments s and s' , let p_s and $p_{s'}$ be the their assigned routes. Let E_l be the set of links located in the closed neighborhood (i.e., containing also link l) of l . There is a *conflict* on link l if s is traveling on that link and s' is traveling on a link $e \in E_l$, simultaneously.

Finally, parameter α weights the risk associated to conflicts. The grater α the higher is the importance of equitably risk spreading with respect to the minimization of the total risk of the plan.

By scheduling the departure time carefully, the risk of each shipment can be reduced, and the safety of each shipment can be ensured.

Combining scheduling and routing analysis generally increases the computational complexity required in treating these components separately. Consequently, even though scheduling and routing decisions are intimately related, their combined analysis has received considerably less attention than is deserved in general.

We have decided to follow a metaheuristic approach, designing a genetic algorithm to solve this problem.

3 Genetic Algorithms

Genetic algorithms (e.g., see [4]) are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. Two characteristic aspects of many traditional algorithms are completely extraneous to the genetic algorithms: the risk that the optimum found is local only and not global, and the dependence from the existence of derives. Likewise, methods of exhaustive character or random search, even not introducing the

two characteristics now quoted, does not reveal also a satisfactory degree of efficiency, because very often the spaces of search are very vast.

An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. Even if a genetic algorithm uses selection and recombination operators to generate new sample points in a search space, which are natural selection laws, there exists a kind of decisive factor which guides the algorithm to recognize the elements to combine in order to design the next generations: that is, their fitness function, which represents the objective function value obtained by the evaluation of the parameters for that particular element.

An evaluation of these chromosomes gives the possibility to allocate reproductive opportunities in such a way that those chromosomes which have a function value better than the others are given more chances to reproduce. The goodness of a solution is typically defined with respect to the current population.

First of all two chromosomes that belong to the current generation are selected (according to their fitness function), then a random cut point is selected and pairs of chromosomes are mated. After this operation, called crossover, the mutation of the chromosomes starts, that is, a random change of the value of one particular parameter of the selected element.

Genetic algorithms require that all the variables of the optimization problem are codified, using an opportune alphabet, in form of string.

4 Implementation

The first step we have to take in order to solve the problem of routing and scheduling using genetic algorithms is to define a suitable coding.

Let n be the number of shipments and let K be the number of dissimilar routes at minimum risk for each shipment. Fig. 1a shows how each chromosome codifies a possible solution of our problem, and Fig. 1b shows that one gene, formed by two alleles, is used to represent the directive of each shipment. The former allele al^0 represents the assigned departure time of the shipment, and the latter al^1 represents the assigned route.

Let al_i^1 be the allele al^1 of the i -th gene (i.e., the i -th shipment), and let $r(al_i^1)$ be the risk associated to the route related to the i -th shipment.

The allele for scheduling (i.e., representing the assigned departure time) uses as alphabet the set of integer numbers in $[-\lambda, \lambda]$, with λ being the maximum number of time slot that a shipment could be anticipated or delayed with respect to the preferred departure time: the value of this allele of the gene at position i indicates the time variation from the preferred departure time of i -th shipment; that is, the departure time dt_i of i -th shipment is given by $dt_i = pdt_i + \delta al_i^0$, where δ is the time slot length (e.g., $\delta = 15$ minutes, $\lambda = 2$).

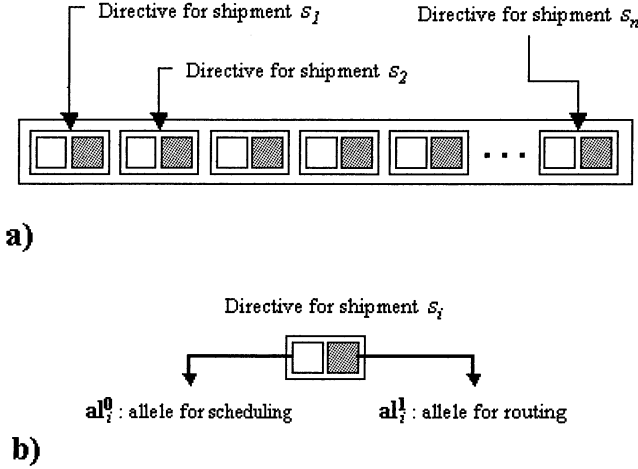


Fig. 1. Genetic representation: (a) chromosome; (b) gene (shipment directive)

The allele for routing uses as alphabet the set of integer numbers in $[1, K]$: the value of this allele of the gene at position i indicates which one of K pre-calculated dissimilar routes at minimum risk is assigned to the i -th shipment.

4.1 Fitness Function

In the evolution theory of the biological systems, the probability of survival of an individual depends on its ability to adaptation to the environment (fitness).

The fitness function $ff(ch)$ is the objective function (1) expressed in terms of a chromosome ch , that is:

$$ff(ch) = \sum_{i=1}^n r(al_i^1) + \alpha \sum_{l \in L} \rho_l c_l(\mathbf{al}^1, \mathbf{al}^0), \quad (3)$$

where \mathbf{al}^0 , \mathbf{al}^1 are the set of alleles of chromosome ch , representing departure times and assigned routes, respectively.

Given the departure time of the shipment s and given the times required to travel on each link of the route assigned to s , it is possible to compute, for each link of that route, the time interval when the link is traversed by s . Let I_l be the set of such time intervals related to link l , computed considering all the shipments. Accordingly, there is a conflict on a link l if there is at least a pair of intersecting time intervals in I_l . Moreover, we also consider possible conflicts on two close links, say links l_1 , l_2 : there is a conflict if there is at

least a pair of intersecting time intervals, the former one belonging to I_{l_1} and the latter one to I_{l_2} . We consider close links that ones being at distance not grater than a given value.

Clearly, the chromosome with minimum fitness function value is the best.

Next the operators reproduction, crossover and mutation are discussed.

4.2 Reproduction

Reproduction is a process in which chromosomes are selected according to their fitness function (ff) value. We have implemented the reproductive operator creating a biased roulette wheel where each chromosome ch in the current population has a slot sized inversely (being a minimization problem) proportional to its ff value and proportionally to its reproduction probability $pr(ch)$.

The turn of the wheel is simulated twice; so two elements are selected.

4.3 Crossover

The second operator is the simple crossover or *one-point* crossover and works as follows: each pair of strings reproduced undergoes a cross-change. In order to do this, an integer position k (cut point) is selected uniformly at random among the strings between position 1 and the string length h less one. The crossover procedure swaps all genes between the selected position k and the end of the string.

We have implemented two other types of crossover:

- *multi-point* crossover: pick M cut points $(p_0, p_1, \dots, p_{m-1})$ at random between 1 and h less one, sort them, and then exchange between the two chromosomes being crossed over the genes in the ranges $[p_0, p_1), [p_2, p_3), \dots$;
- *uniform* crossover: for each gene position in the two chromosomes being crossed over, flip a coin and exchange the genes if the coin comes up heads.

4.4 Mutation

After the determination of the two new elements with the crossover, operates the mutation function, which plays a secondary role with respect to reproduction and crossover operators. A gene selected for mutation is replaced by a random value between lower and upper bounds of the alphabet of the gene.

This procedure gives a random alteration of the value of the genes that forms the chromosomes, with small probability, and guarantees the exploration of some regions in the search space not observed yet.

Once we have processed the population through the three fundamental operators, a new generation is created. The new fitness values are calculating and the best chromosome is tagged, that is, the one with the best fitness value.

5 Preliminary Results and Conclusions

The genetic algorithm implementation was written in Java and the calculations were carried out on a PC with a 450 MHz Pentium II processor.

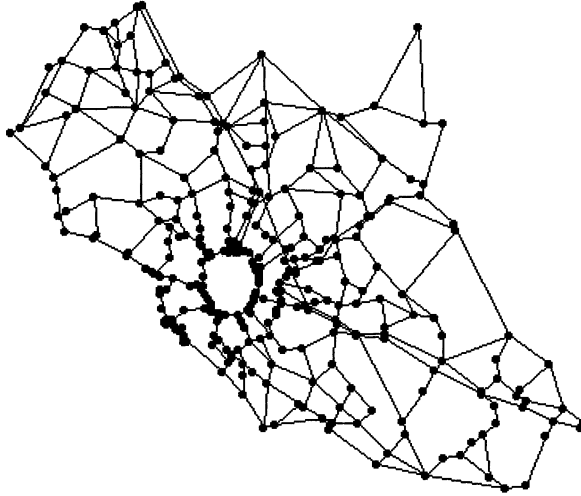


Fig. 2. The regional road network for the experiments

In order to test the performance, calculations have been performed on a realistic regional road network, shown in Fig. 2, with 311 nodes and 441 links.

We have fixed 24 origin-destination pair and 5 shipments for pair; then chromosomes have length 120, being the total number of shipments. Every shipment has a randomly generated preferred departure time in the interval [07:00 AM, 08:00 AM]. The time slot length δ is 15 minutes and $\lambda = 2$, then $pdt_i - 2\delta \leq dt_i \leq pdt_i + 2\delta$, that is $pdt_i - 30min \leq dt_i \leq pdt_i + 30min$.

For each origin-destination pair we have precalculated 3 dissimilar routes at minimum risk using the Iterative Penalty Method [3].

Population size is a factor that effects the GA performance. Increasing the population means a longer computational time; on the other hand, if the population size decreases, the same happens to the accuracy of the solution, because of reduced variation of chromosomes. In GA design there must be a balance between the generation numbers and population size. Evaluation of chromosomes and fitness calculations are the most time consuming parts of a GA. If the evaluation operation is reduced, GA process will work faster and this can be achieved by reducing the population size and numbers of generation needed to reach a solution. Parameters in the proposed GA are set as: population size $N = 20$, and maximal search generation $I_{max} = 200$.

In the following tables we report results of a set of experiments performed with different combinations of crossover and mutation probability. In particular we have considered different crossover probabilities ($cp = 0.50, 0.70, 0.85, 1.00$) and mutation probabilities ($mp = 0.001, 0.010, 0.100, 0.200$) and 10 trials are done for each combination. Results show that the best ff value is obtained with ($cp = 0.85, mp = 0.200$), and ($cp = 0.70, mp = 0.100$), using one-point crossover and uniform crossover, respectively.

Table 1. Best fitness values for different combinations of crossover and mutation probability using one-point crossover

mp	$cp = 0.50$	0.70	0.85	1.00
0.001	9104974.2	8985228.2	8959705.4	8823420.6
0.010	8798133.8	8737627.4	8908269.0	8877275.6
0.100	8833818.4	8806757.2	8722281.4	8736266.3
0.200	8644340.3	8579511.3	<u>8576728.0</u>	8600005.0

Table 2. Best fitness values for different combinations of crossover and mutation probability using uniform crossover

mp	$cp = 0.50$	0.70	0.85	1.00
0.001	9173154.3	8982624.2	8851794.2	8633443.3
0.010	8727306.6	8873224.6	8731722.6	8755767.0
0.100	8620333.4	<u>8588053.8</u>	8679612.2	8785719.3
0.200	8703173.3	8678770.0	8725388.6	8785198.0

Further developments concern the algorithm test considering different methods to find dissimilar paths at minimum risk and a comparison of the performance with an other metaheuristic approach based on Tabu Search.

References

1. V. Akgn, E. Erkut, R. Batta, On finding dissimilar paths, *European Journal of Operational Research* 121, (2000) 232-246.
2. E. Erkut, V. Verter, Modeling of transport risk for hazardous materials, *Operations Research* 46 (5), (1998) 625-642.
3. P.E. Johnson, D.S. Joy, D.B. Clarke and J.M. Jacobi, "HIGHWAY 3.01, An Enhanced Highway Routing Model: Program, Description, Methodology, and Revised User's Manual", Oak Ridge National Laboratory, ORNL/TM-12124, Oak Ridge, TN, (1992).
4. David E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, (1989), Reading, MA.

Personal- und Fahrzeugeinsatzplanung in der Müllentsorgung

Joachim R. Daduna

FHW Berlin; Badenische Straße 50 – 51, D – 10825 Berlin

Zusammenfassung Die Diskussion um eine leistungsfähige und auch kostengünstige Entsorgung der anfallenden Müllmengen hat sich in den vergangenen Jahren erheblich verschärft. Dies bezieht sich (u.a.) auch auf die Hausmüllentsorgung, die durch die unmittelbare Betroffenheit der privaten Haushalte eine (kommunal-)politische Dimension beinhaltet. Diese Situation erzwingt eine stärkere Beachtung betriebswirtschaftlicher Zielsetzungen, u.a. in der Form von betriebsinternen Kostensenkungsmaßnahmen. Ein wesentlicher Aspekt ist in diesem Zusammenhang die Verbesserung der Tourenplanung sowie der Personaleinsatzplanung in der Müllentsorgung. Ein möglicher Lösungsansatz für diese Problemstellung wird in den Grundzügen skizziert. Außerdem werden die mit einer Realisierung verbundenen Randbedingungen beschrieben sowie notwendige Veränderungsprozesse aufgezeigt.

1 Anforderungen an die Entsorgungslogistik

Die *Müllentsorgung* erweist sich für die *kommunalen Gebietskörperschaften* zunehmend als ein politisches und zum Teil auch als ein finanzielles Problem. Ein wesentlicher Bereich ist hierbei das Sammeln des anfallenden *Hausmülls* und der Transport zu Entsorgungseinrichtungen, der aufgrund einer Reihe gesetzlicher Regelungen in den vergangenen Jahren (u.a. durch das *Kreislaufwirtschafts- und Abfallgesetz* (KrW-/AbfG)) erheblich komplexer geworden ist. Mit den ab Juni 2005 wirksam werdenden Regelungen für die Entsorgung (mit der verbindlichen Festschreibung einer geeigneten Verwertung) ergeben sich zusätzliche Anforderungen, u.a. auch an die logistischen Abläufe, für die geeignete Lösungen zu ermitteln sind.

Ein Teilbereich, der nicht genutzte Effizienzpotentiale aufweist, ist der (operative) *Personal- und Fahrzeugeinsatz* bei Anwendung des Umleerverfahrens in der Hausmüllentsorgung. Die hier auftretenden Abläufe sind bisher weitgehend als in sich geschlossene Prozesse verstanden worden, bei denen Fahrzeug und Fahrer sowie die Ladecrew eine (nicht trennbare) Arbeitseinheit bilden. In der Grundstruktur bestehen diese Prozesse aus einer Einsetzfahrt (ausgehend vom Depot), den Sammeltouren sowie den Fahrten zum Entladen, und einer Aus-

setzungsfahrt (zurück in das Depot) (s. Abb.1.) Aufgrund von Veränderungen in den räumlichen Randbedingungen, u.a. durch zunehmende Entfernungen zwischen den Aufkommensgebieten des Hausmülls und den verschiedenen Zielorten der Sammelfahrzeuge (Umschlageinrichtungen, Deponien, Recyclinganlagen, etc.), ergibt sich ein Ansteigen der *unproduktiven Zeitanteile* in den Arbeitsabläufen und damit auch der anfallenden Kosten.

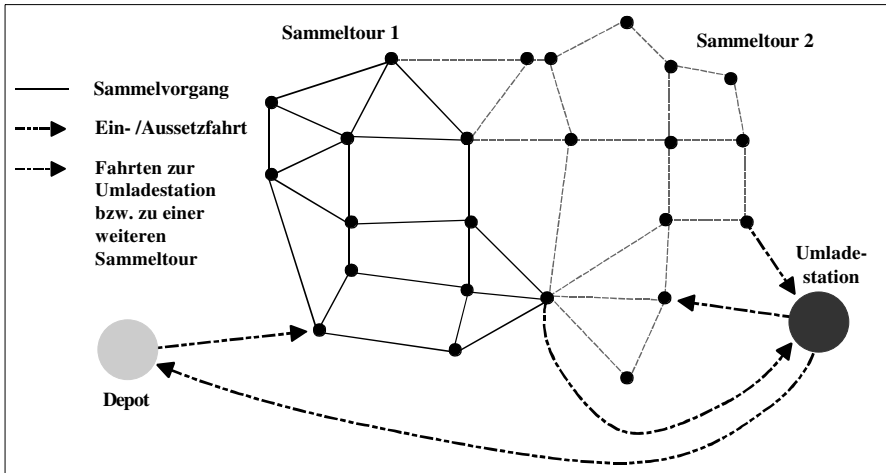


Abb. 1: Grundstruktur der Abläufe bei der Tourenbildung in der Müllentsorgung

Kostensteigerungen in diesem Marktsegment können in der Regel nur bedingt an den Kunden weitergegeben werden, denn Gebührenerhöhung bei öffentlichen Dienstleistungen sind nicht nur kaufmännische Entscheidungen, sondern sie stellen auch ein (kommunal-)politisches Problem dar. Dies führt zu einem erheblichen Druck auf die betroffenen Entsorgungsunternehmen, verstärkt Kostensenkungsmaßnahmen zur Erhöhung der Wirtschaftlichkeit durchzusetzen. Um diesen Forderungen gerecht zu werden, müssen neue Wege gesucht werden, mit denen auch ein Aufbrechen bestehender Strukturen verbunden sein kann.

2 Problemstellung

Eine mögliche Vorgehensweise zur Verbesserung der Wirtschaftlichkeit besteht in einer Veränderung der Entsorgungsabläufe. Der kritische Punkt liegt (u.a.) in den nicht produktiven Arbeitszeitanteilen der Ladecrew während der Fahrten zu einer Umschlagstation bzw. während der Aussetzfahrt bei größeren Entfernungen zwischen Depot und Entladeort. Grundlage für einen Lösungsansatz ist das Aufbrechen der *Arbeitseinheit* Fahrzeug / Fahrer / Ladecrew, mit dem Ziel, das verfügbare Ladepersonal effizienter einzusetzen. So ist in der (operativen) Personal- und Fahrzeugeinsatzplanung anzustreben, dass eine Ladecrew bei längeren Fahrten

am Abschluss einer Sammeltour, (zum Beispiel) zu einer Umschlagstation, nicht (untätig) auf dem Fahrzeug verbleibt, sondern an einem Ablösepunkt einem anderen (aktiven) Fahrzeug zugeordnet wird (s. Abb. 2).

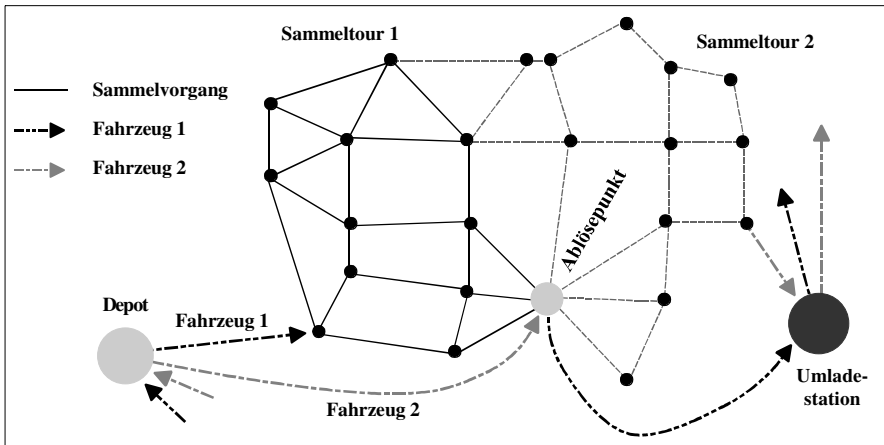


Abb. 2: Ablaufstruktur bei Fahrzeugwechsel der Ladecrew

Hier bieten sich zwei Lösungsansätze an, zum einen ein zweistufiger Ansatz, bei dem zunächst eine Tourenplanung für die Fahrzeuge (und Fahrer) erfolgt und anschließend eine Zuordnung des Ladecrews zu den zu leistenden (produktiven) Tourabschnitten, sowie zum anderen ein simultaner Ansatz, der allerdings eine deutlich komplexere Struktur aufweist. In den folgenden Ausführungen wird, auch mit Blick auf die Komplexität und das betriebliche Handling, auf den zweistufigen Ansatz eingegangen, wobei in den Grundstrukturen von einem Ein-Depot-Problem ausgegangen wird.

3 Lösungskonzept

Die Vorgehensweise basiert auf zwei Planungsschritten, die mit (vorhandenen) quantitativen Verfahren bearbeitet werden können. So ist in einer ersten Stufe ein *Capacitated Arc Routing-Problem* (CARP-Problem) (s. hierzu u.a. [1], [3]), [5] für den Einsatz von Fahrzeug und Fahrer zu lösen. Dieses liefert, unter Berücksichtigung der Arbeitszeiten der Fahrer, die zu erbringenden Leistungen (Touren), die sich zusammensetzen aus der Einsetzfahrt, den (produktiven) Sammelvorgängen und gegebenenfalls mehreren Fahrten zwischen Sammelgebiet und (zum Beispiel) einer Umschlagstation sowie der Aussetzfahrt am Ende der Tour. Bei einem Schichtbetrieb kann, mit Blick auf einen effizienten Einsatz der Fahrzeuge, die zulässige Länge einer Tour auch auf die tägliche Betriebsdauer ausgeweitet werden, was allerdings voraussetzt, dass ein Fahrerwechsel innerhalb einer Tour ablaufmässig realisierbar ist.

Ausgehend von dieser *Routing-Lösung* werden alle *produktiven Tourenabschnitte* erfasst, die mit Ladecrews zu besetzen sind. Diese bilden den wesentlichen Teil der Datengrundlagen für die Festlegung der durch die Ladecrews zu erbringenden Arbeitsleistungen (Dienste). Die hier zugrunde liegende Problemstellung der zweiten Stufe lässt sich als (klassisches) *Duty-Scheduling-Problem* formulieren, das mit verfügbaren Verfahren effizient gelöst werden kann (s. u.a. [2], [6]). Abbildung 3 zeigt (in einem Ausschnitt) das Ergebnis einer solchen Einsatzplanung im Vergleich zu den vorhandenen Vorgehensweisen in einer Gantt-Darstellung.

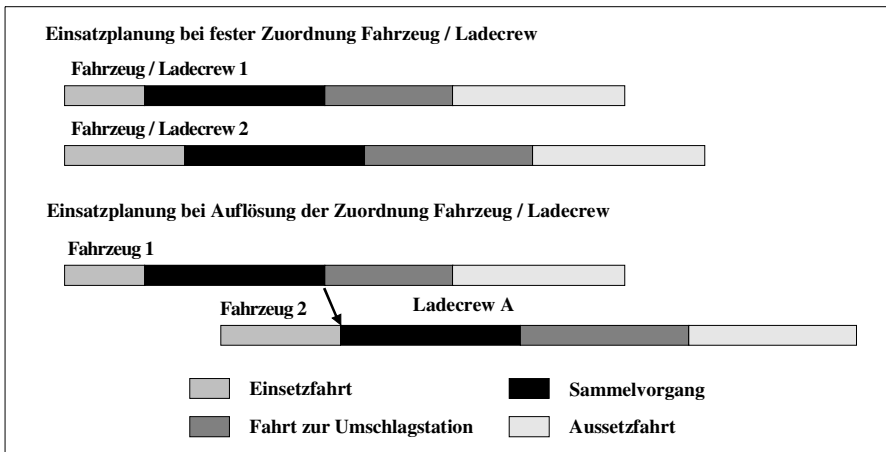


Abb. 3: Beispiel Einsatzplanung (Ausschnitt)

Allerdings bestehen bei dieser Vorgehensweise auch Probleme, da durch ein zweistufiges Vorgehen auf der algorithmischen Ebene keine unmittelbare Verknüpfung bzw. Rückkopplung zwischen den beiden Stufen hergestellt werden kann. Da (u.a.) räumliche und zeitliche Interdependenzen zwischen dem Verlauf der einzelnen Touren und der Einsatzplanung der erforderlichen Crews bestehen, müssen die automatisiert ermittelten Lösungen überarbeitet werden. In einem ersten Schritt können hierbei geeignete Verbesserungsverfahren zur Anwendung kommen, die eine inhaltliche Verknüpfung zwischen den beiden (Teil-)Problemstellungen herstellen. Außerdem ist in einem zweiten Schritt die Möglichkeit *interaktiver* Eingriffe des Planungspersonals vorzusehen, um aus betrieblicher Sicht erforderlichen Anpassungen bei der Touren- und Dienstebildung vornehmen zu können.

Eine interaktive Anpassung der (weitgehend) automatisiert erstellten Touren- und Dienstpläne ist zwingend notwendig, da in der Regel eine Reihe betrieblicher Einzelfallregelungen, denen häufig auch eine ausreichende Konsistenz fehlt, bei der Modellbildung keine Berücksichtigung finden können. Dies ist allerdings nicht nur aus Effizienzüberlegungen erforderlich, sondern auch um die Akzeptanz des

betroffenen Personal bei der Realisierung einer solchen Vorgehensweise zu erreichen.

Ein weiterer wichtiger Punkt sind die sich ergebenden Einflüsse auf die Arbeitszeitregelungen. Da der Fahrzeugeinsatz mit der Einteilung der Ladecrews synchronisiert werden muss, ergibt sich zwangsläufig eine Flexibilisierung der Arbeitszeiten, d.h. die bisher genutzten statischen Schichtmodelle können nicht mehr zur Anwendung kommen. Grundlage für ein flexibles Arbeitszeitkonzept kann die Einsatzplanung bei der Leistungserstellung im Bereich des öffentlichen Personennahverkehrs sein, wo im operativen Bereich traditionell eine bedarfsangepasste Festlegung der Arbeitszeiten des Fahrpersonals erfolgt

Neben den Veränderungen bei den (operativen) Planungsabläufen ergeben sich bei einem derartigen Konzept weitere Anforderungen im Dispatching, da sich der Koordinationsaufwand zwangsläufig erhöhen wird. So ist ein effizientes *Informationsmanagement* erforderlich (s. u.a.[4]), um auf der Basis online erfasster Betriebsdaten eine leistungsfähige *Steuerung* und *Überwachung* der Abläufe gewährleisten zu können. Dies bezieht sich in erster Linie auf eine zeitgenaue Bereitstellung der Ladecrews, die bei Dienstbeginn bzw. nach Abschluss eines Sammelvorgangs entweder direkt von einem Leerfahrzeug an einem definierten Punkten übernommen werden oder aber mit einem anderen Fahrzeug zu dem nächsten Einsatzpunkt (bzw. bei Dienstende zurück zum Depot) transportiert werden (Taxi-Trip). Aufgrund der verfügbaren *Informations-* und *Kommunikationstechnologie* (Standort Erfassung der Fahrzeuge über satelliten-gestützte Ortung, Online-Erfassung von Fahrzeugdaten, etc.) sind die notwendigen (technischen) Voraussetzungen für das erforderliche Dispatching gegeben.

4 Zusammenfassung und Ausblick

Die notwendigen Veränderungen in der (kommunalen) Entsorgungswirtschaft erfordern in verschiedenen Bereichen ein tiefgreifendes Umdenken in der Gestaltung der logistischen Prozesse. Mögliche Ansätze bestehen, neben Maßnahmen auf der (fahrzeug-)technischen Ebene, in einer effizienteren Touren- und Personaleinsatzplanung, die mit vorhanden quantitativen Methoden realisiert werden kann. Verbunden hiermit sind allerdings auch verschiedene Maßnahmen im organisatorischen Bereich, die zwangsläufig erhebliche Veränderungen der vorhandenen (betrieblichen) Strukturen zur Folge haben werden.

Im Vordergrund steht hierbei, neben der Aufhebung der festen Zuordnung von Fahrern und Ladecrews, die Flexibilisierung der Arbeitszeiten. Außerdem ergibt sich im operativen Bereich für den Ladecrews eine Erhöhung der produktiven Arbeitszeitanteile, die zwangsläufig zu Personaleinsparungen führen. Allerdings entsteht im Bereich der Planung und des Dispatchings auch ein zusätzlicher Personalbedarf, allerdings in einem geringeren Umfang, da es hier zu einem deutlichen Ansteigen des Aufgabenumfangs kommt.

Bei einer Durchsetzung derartiger innerbetrieblicher Veränderungen muss mit Sicherheit in vielen Fällen mit erheblichen Widerständen gerechnet werden, da

insbesondere in den durch kommunale Gebietskörperschaften beherrschten Unternehmen die Akzeptanz wettbewerblicher Strukturen (weitgehend) fehlt. Außerdem zeigen sich immer wieder Bestrebungen nach einer möglichst weitgehenden Besitzstandswahrung, die notwendigen Änderungen entgegenstehen. Um allerdings in einem deregulierten Markt zukünftig bestehen zu können, müssen die möglichen Verbesserungsmaßnahmen umgesetzt werden, um langfristig die Konkurrenzfähigkeit am Markt gewährleisten zu können.

Literatur

1. Assad, A.A. / Golden, B.L. (1995): Arc routing methods and applications. in: Ball, M.O. / Magnanti, T.L. / Monma, C.L. / Nemhauser, G.L. (eds.): Network routing. (North-Holland) Amsterdam, 375 - 483
2. Banihashemi, M. / Haghani, A. (2001): A new model for the mass transit crew scheduling problem. in: Voß, S. / Daduna, J.R. (eds.): Computer-aided scheduling of public transport. (Springer) Berlin et al., 1 - 15
3. Bodin, L.D. / Golden, B.L. / Assad, A.A. / Ball, M.O. (1983): Routing and scheduling of crews. in: Computers & Operations Research 10, 63 - 212
4. Daduna, J.R. / Voß, S. (2000): Informationsmanagement im Verkehr. in: Daduna, J.R. / Voß, S. (Hrsg.): Informationsmanagement im Verkehr. (Physica) Heidelberg, 1 - 21
5. Hertz, A. / Mittaz, M. (2001): A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. in: Transportation Science 35, 425 - 434
6. Mingozzi, A. / Boschetti, M.A. / Ricciardelli, S. / Bianco, L. (1999): A set partitioning approach to the crew scheduling problem. in: Operations Research 47, 873 - 888

Modelling of Complex Costs and Rules in a Crew Pairing Column Generator

Rastislav Galia and Curt Hjorring

Carmen Systems AB, Odinsgatan 9, Sweden
rastislav.galia@carmensystems.com
curt.hjorring@carmensystems.com

Abstract. Crew pairing, the creation of anonymous lines of work, is a crucial part of the crew airline planning process. Column generation with shortest path pricing subproblem provides high quality solutions. In its basic form, the pricing subproblem relies on assumptions, such as additivity of the cost function and constraint contributions. However, it is not possible to be assume that these requirements are satisfied, particularly if the pairing system gives the user control over problem formulation and maintenance.

Solutions to these challenges are proposed, based on proper granularity of the subproblem, a k -shortest paths based pricer and the application of resources to model nonadditive costs. Furthermore, a label-merging technique provides significant performance improvements.

1 The Pairing Problem

The pairing problem consists of creating anonymous sequences of flight legs (pairings), so that the coverage requirement of each leg is fulfilled. Each pairing is required to start and end at the same homebase airport and satisfy legality constraints (rules). The rules are often of a complex structure and vary from airline to airline.

Pairing costs consists mainly of salary, per diem compensation, hotel costs and artificial penalties introduced to reward some characteristics of the pairings. The resulting cost function is in general nonadditive. The optimisation problem is to minimise the sum of all the pairing costs.

1.1 Approaches Survey

The pairing problem can be formulated as a set-covering problem. Solving this problem to optimality requires explicit enumeration of all legal pairings, which is computationally very expensive due to the very large number of legal pairings.

The problem can instead be solved approximately by an iterative approach. In each iteration the previous solution is used as a starting point. New pairings that are similar to the pairings in the previous solution are generated and the iteration is concluded by solving a set-covering IP problem with pairings from the previous solution and the newly generated ones.

Various heuristic strategies are used in the generation step, for example time-windowing. This is known as the *generate and optimise* approach. Further details can be found in [2].

Another widely used technique, column generation, is based on the idea of solving the LP-relaxation of the original problem. New pairings with negative reduced costs are generated by solving the pricing subproblem. If the *pricing subproblem* is solved optimally, this method will find the optimal solution to the LP-relaxation.

An early application of column generation to the pairing problem is due to [7], in which it is shown that columns can be priced by solving a shortest path problem on a network with arcs representing legs and overnight connections. In the airline industry the sequence of legs flown in a working day is known as a *duty period*, and for many rules the legality of a single duty period is independent of preceding or succeeding duties. This additional structure is exploited in [5] where a duty period network is formed in which nodes are duty periods with state information, and arcs represent legal overnights.

Results for flight and duty based implementations are presented in [8]. In both versions, resources (labels) are added to each node to track legality conditions, and a *resource constrained shortest path problem* is solved. Comparisons between the two versions show that flight based implementation generally spends more time in the pricing routine. However, the duty version cannot solve as large problems as the flight version due to memory limitations.

The problem of the size of the duty network is addressed in [4]. A relaxed duty network with a significantly smaller number of arcs is introduced. This has a positive impact on memory consumption as well as the time spent in the pricing routine. A refinement scheme for the duty network (partial reversing of the relaxation) is proposed.

In the next sections we will build on top of the framework proposed by [4] by extending the class of modelled rules and cost functions. In section 2 we show how some results can be achieved by the proper design of the network topology. Some more complex rules and costs require modification of the pricing subproblem solver, as shown in section 3.

2 The Solution Method

2.1 Rules and Costs, their Structure and Evaluation

Carmen Systems provides a rule modelling language (Rave) that permits planners to easily add and modify the rules and the cost function. Rave is a black-box system, able to evaluate expressions on chained legs and deciding whether a chain is legal with respect to the rules. However, it does not expose the internal details of the evaluation process to the rest of the system.

The structure of the rules and costs has significant impact on the pricing process. For modelling purposes the sequence of legs is usually grouped into

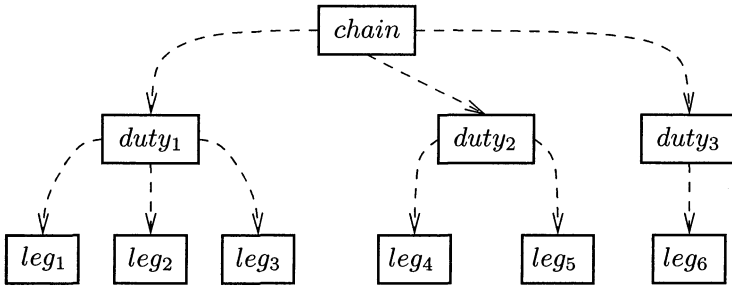


Fig. 1. Hierarchy of chain levels

the hierarchy of levels, as shown in the figure 1. In the pairing problem, there is usually only one intermediate level of *duties* between the atomic level of the legs and the top-most level.

An important concept is the range of objects that evaluated expressions depend on. For example, an expression returning “block time of current duty” has range of one duty. An expression determining the length of the pairing in terms of days has chain range.

2.2 Network Model

The pricing routine, as proposed by [7], finds a pairing with negative reduced cost as a shortest path in the network. This basic approach relies on the additivity of the cost function and the legality of the pairings corresponding to the path in the network.

These assumptions are very seldom satisfied entirely, so [4] proposes to introduce a k -shortest path (k -SP) based pricing subproblem solver. In this design, the pricing routine works by first calculating the shortest path tree for the network. This returns the cheapest path in the network. If the cost is nonnegative, there are no attractive pairings, and the pricing routine terminates.

If the path has a negative reduced cost estimate (which is obtained by summing up the arc costs), the rule system is called to see if the corresponding pairing is legal, and if it also has negative reduced cost. If so the pairing is added to the LP, and the pricing routine can either terminate, or continue producing attractive pairings. If the pairing is illegal, or does not have negative reduced cost, then the pricing routine finds the next shortest path, and repeats the tests. This continues until one of the above termination criteria is satisfied, or all paths are enumerated.

As mentioned in [4] and [5], the fact that the majority of the pairing rules have range of one or two duties can be exploited to design an efficient pricing routine. Approximate additivity of the cost function with respect to the duties and duty-duty connections reduces the frequency and magnitude of cost estimation errors as well.

2.3 Network Topology

The network is designed to model completeness criteria (a pairing has to start and end at the same base), maximum length of the pairing in calendar days and nonadditive penalty dependent on the length of the pairing.

This is illustrated in figure 2. There are multiple source and sink nodes, one for each calendar day and homebase combination. A separate k -SP pricing is performed for each starting node, and the search is restricted to paths leading to the corresponding homebase within a specified number of days. This way the maximum length of the pairing and the completeness rule is satisfied. Arcs between the sink nodes and the “master sink” node model calendar length penalty.

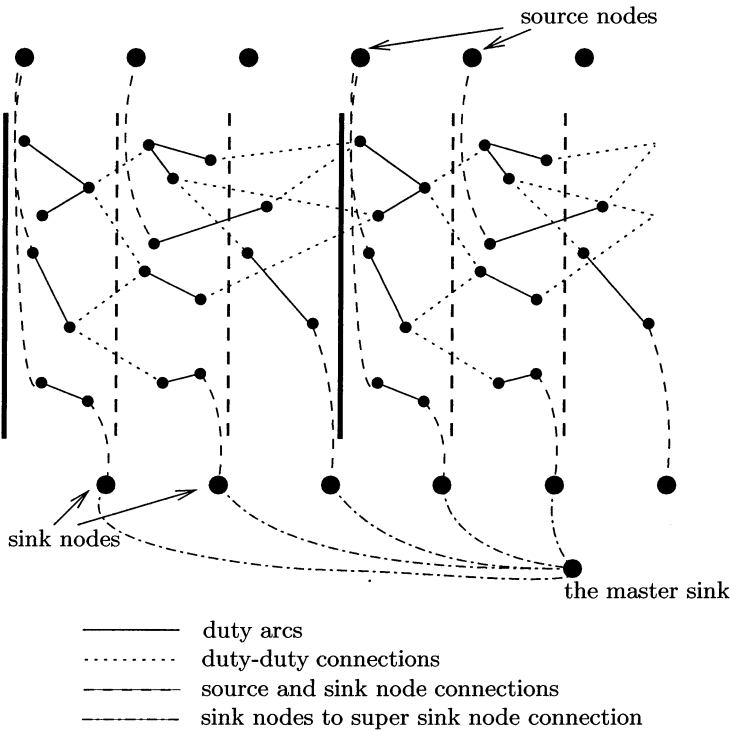


Fig. 2. Overview of the network model

3 Extension to the Network Model

The basic network model gives the foundation for a successful combination of column generation with a black-box rule system. However, for some classes of

problems the cost estimation errors and the rule failures can severely impact the performance or the solution quality. Here we present how one such class of problems can be modelled in the network and how this change affects the k -SP routine.

3.1 Resource Dependent Cost Elements

The resource constrained shortest path problem has traditionally been used as a subproblem in applications of column generation to scheduling problems. The technique can be extended to model some classes of nonadditive costs as well.

Let us assume that cost of a pairing can be written in the form $c(p) = c_a(p) + c_n(r(p))$, where $c_a(p)$ is additive with respect to the duties and duty connections. $c_n(r(p))$ is the nonadditive part dependent on additive resource vector $r(p)$.

Not every nonadditive cost can be modelled using resources. The resource must be a additive function. Furthermore, the nonadditive part of the cost c_n is required to be nonnegative and nondecreasing with respect to the value of the individual resources.

In practise, c_n may depend also on some other characteristics of the path that can be determined directly from the network, such as starting day of the pairing (given by path's source node) or the length in calendar days. Some examples of costs modelled by resources:

- Penalise pairings for which the flight time exceeds 480 minutes by constant value of 4000 cost units.
- Penalise pairings for number of the short night stops (< 6 hours). One short night-stop is not penalised, two or more short night-stops are penalised by 500.

Modification of the Network and the k -SP Algorithm for Use with the Resources

The network model is modified in such a way that additive part \bar{c}_a of the reduced cost is kept on the arcs instead of the original reduced cost \bar{c} . In addition, arc *resource consumption* vector $r(p)$ is stored in the graph as well.

The pricing subproblem consists of finding any attractive pairing (with negative reduced cost $\bar{c}(p) = c(p) - \pi A$). However, it is preferable to find several attractive pairings in one pricing iteration.

If there are no resources, the k -SP algorithm achieves this by enumerating all paths in ascending order with respect to the reduced cost. It stops after encountering the first non-attractive path, at which point all attractive paths have been found.

With resources, a nonadditive pricing subproblem needs to be solved and the k -SP algorithm can not guarantee that the generated paths are ordered

by their reduced cost, as illustrated by the example in figure 3. In the example pricing subproblem the cost function is the sum of a additive cost and a nonadditive part c_n depending on one resource. Let us assume, that $c_n(r(p)) = 0$ if $r(p) < 70$ and $c_n(r(p)) = -5$ otherwise.

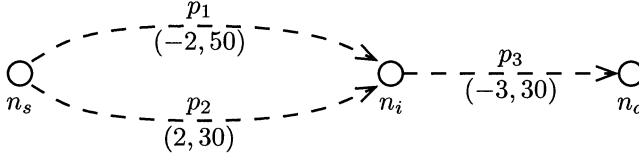


Fig. 3. Impact of the resources on the k -SP algorithm

Even though the cost of the path p_1 is $-2 + c_n(50) = -2$ and cost of the path p_2 is $2 + c_n(30) = 2$, we cannot say that p_1 is definitely better than p_2 , since $\bar{c}(p_1 \circ p_3) = -5 + c_n(80) = 0 > \bar{c}(p_2 \circ p_3) = -1 + c_n(60) = -1$.

Attractive paths could be found by enumerating all paths from the source to the sink node. However, this process is obviously very inefficient. In order to reduce the number of generated paths, we search for the set of *Pareto-optimal* paths from the source to the sink node.

Definition 1. The path p is said to be *dominated* by path p' , if $\bar{c}_a(p) \geq \bar{c}_a(p')$ and $r_j(p) \geq r_j(p')$ for every resource r_j .

Definition 2. Let $P(n_s, n)$ be set of all paths from the source node to node n . A path is called *Pareto-optimal*, if it is not dominated by any path in $P(n_s, n)$.

If the set $P(n_s, n_d)$ of paths from the source node n_s to the sink node n_d contains at least one attractive path, then it contains also at least one attractive Pareto-optimal path. This means that the pricing subproblem is guaranteed to be solved by this approach.

A modification of the algorithm due to [3] is used to find the set of the Pareto-optimal paths. An important property of the Pareto-optimality is the “principle of optimality”, meaning that the subpath to an intermediate node in the network, which is not Pareto-optimal, cannot be part of a Pareto-optimal complete path to the sink node. This allows us to discard dominated subpaths in the early stages of the network and to significantly reduce the computational effort.

Pareto-optimal subpaths to each node are enumerated in increasing lexicographic order with respect to \bar{c}_a and vector of resources $r(p)$. The k -SP terminates after encountering the first path to sink node with $\bar{c}_a \geq 0$.

Label Merging

Due to frequent rule failures and underestimates, the k -SP routine can spend very long time in certain parts of the network until it finds the attractive pair-

ings. These problems might be resolved in later stages thanks to a refinement of the duty network. However, it is preferable to focus the search effort to those parts of the network, which are more likely to produce attractive legal rotations.

A simple adaptive technique can be used to regulate the number of generated non-dominated paths in the network by over-restricting the dominance with the objective to dominate a larger number of paths which are represented by a “merged label (path)”.

Path p is said to be dominated by path p' if $\bar{c}_a(p) + \kappa_a(n_s) \geq \bar{c}_a(p')$ and $r^{(j)}(p) + \kappa^{(j)}(n_s) \geq r^{(j)}(p')$ for each resource $r^{(j)}$, where $\kappa_a(n_s), \kappa^{(j)}(n_s)$ are parameters depending on the current source node n_s . Parameters $\kappa_a(n_s)$ and $\kappa^{(j)}(n_s)$ are adaptively changed after every pricing iteration, depending on the time spent in a pricing routine for source node n_s .

4 Results

The extensions to the network model and pricing subproblem solver mentioned in section 3 were successfully used in a commercial production system developed by Carmen Systems.

The improved modelling capabilities have been tested on several problem instances of our clients and we observed significant performance improvements. Figure 4 depicts the effect of using the resource modelling and label merging on a pairing problem from KLM. The graph shows that using both new features in the pricer can reduce runtimes by up to 80% on some examples.

5 Summary

By extending the duty-network approach in [4] we have developed a flexible column generator system able to handle wide variety of cost and rule structures while maintaining the black-box paradigm of flexible rule modelling language. In order to better support this flexibility, more sophisticated network algorithms employing resource modelling and label merging were used for solving the pricing subproblem.

Acknowledgements

We would like to thank our colleagues Fredrik Engel, Lennart Bengtsson, Tomas Gustaffson, Johan Ivarsson, Hamid Karraziha and Stefan Karisch who have contributed to this work.

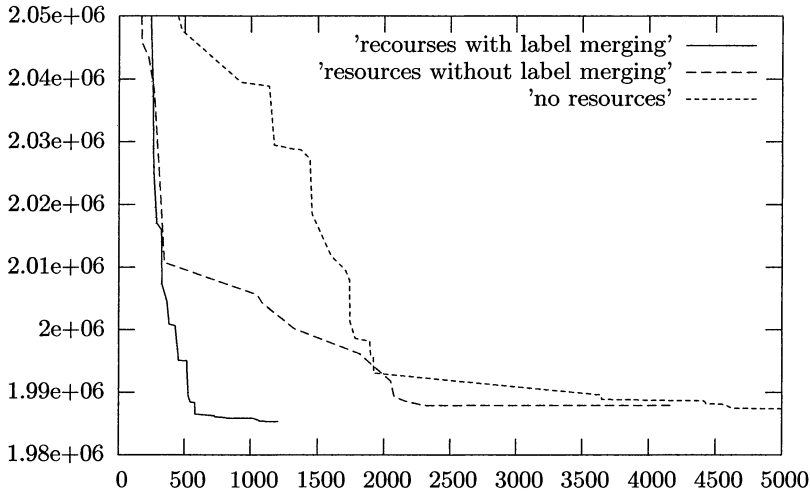


Fig. 4. Behaviour of column generator resource modelling

References

1. R. Anbil, J. J. Forrest, and W. R. Pulleyblank. Column generation and the airline crew pairing problem. *Documenta Mathematica*, Extra Volume ICM III:677–686, 1998.
2. E. Andersson, E. Housos, N. Kohl, and D. Wedelin. Crew pairing optimization. In G. Yu, editor, *Operations Research in Airline Industry*, pages 228–258. Kluwer Academic Publishers, Norwell, MA, 1998.
3. J. A. Azevedo, E. Q. V. Martins. An algorithm for the multiobjective shortest path problem on acyclic networks. *Investigação Operacional*, Vol. 11, pages 52–69, 1991.
4. C. Hjorring and J. Hansen. Column generation with a rule modelling language. In *Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand*, pages 133–142, Hamilton, New Zealand, December 10–11, 1999.
5. S. Lavoie, M. Minoux, and E. Odier. A new approach of crew pairing problems by column generation and application to air transport. *European Journal of Operations Research*, pages 45–58, 1988.
6. E. Martins and J. Santos. *A new shortest paths ranking algorithm*. *Investigação Operacional*, 20(1):47–62, 2000.
7. M. Minoux. Column generation techniques in combinatorial optimization: A new application to crew pairing problems. In *XXIVth AGIFORS Symposium*, 1984.
8. P. H. Vance, A. Atamtürk, C. Barnhart, E. Gelman, E. L. Johnson, A. Krishan, D. Mahidhara, G. L. Newhauser, and R. Rebello. A heuristic branch-and-price approach for the airline crew pairing problem. Technical report, School of Industrial and Systems Engineering, Georgia Tech., June 1997.

Convexification of the Traffic Equilibrium Problem with Social Marginal Cost Tolls

Per Olov Lindberg¹ and Leonid Engelson²

¹ Linköping University, Dept. of Mathematics, SE-581 83 Linköping

² Royal Institute of Technology, Dept. of Infrastructure, SE-100 44 Stockholm
e-mails: polin@mai.liu.se, lee@infra.kth.se

Abstract. In an earlier paper, we have demonstrated that traffic equilibria under social marginal cost tolls can be computed as local optima of a nonconvex optimization problem. The nonconvexity of this problem implies in particular that linearizations, e.g. in the Frank-Wolfe method, do not give underestimates of the optimal value. In this paper we derive the convex hull of nonconvex arc cost functions of BPR type. These convexifications can be used to get underestimates of the optimal value, or to get better search directions in the initial phase of the Frank-Wolfe method. Computational results for the Sioux Falls and Stockholm networks are reported

1 Overview

Traffic in large cities is today a major problem for society. It has become a common standpoint among transportation planners that it is necessary to charge some kind of fees, congestion tolls, for the usage of the crowded streets. Classical social marginal cost pricing theory for road traffic ([1], Ch. 4) states that if each user is charged a toll equal to the total value of time loss incurred on other users of the network, this will induce an equilibrium that is system optimal (i.e. with minimal total travel time), assuming that all users have a fixed and identical time value.

However, different traveller groups typically have radically different time values. Therefore, computed tolled equilibria need to recognize the variation of time values between user groups, thus leading to multi-class user equilibria, (e.g., [2], [8],[4]).

Introducing tolls in this multi-class setting, one would like the tolls to induce an equilibrium that minimizes not the total travel time but rather the total value, V , of travel time as perceived by the users.

In previous papers ([3], [4]) we have shown, that introducing social marginal cost (SMC) tolls will lead to equilibria that do not necessarily minimize V . However, they come close in that they turn out to be points fulfilling necessary conditions for local minima of V (over the set of feasible flows). However, V turns out to be nonconvex, implying that there may be many such equilibria.

These equilibria would typically be determined by attempting to minimize V , e.g. by the Frank-Wolfe method. The nonconvexity of V , however, implies

that one cannot compute lower bounds to the objective by linearizations, as for convex problems. Hence one has no measure of the quality of an achieved solution.

In the current paper we partly overcome this problem by constructing the convex hull, $\text{conv}V$ of V . Minimizing $\text{conv}V$, or using classical Frank-Wolfe bounds for $\text{conv}V$, will give us lower bounds for V .

Moreover, using $\text{conv}V$, rather than V , in the initial phases of the minimization may be favorable, compared to pure Frank-Wolfe iterations.

2 Multi-Class User Equilibria

Consider a road network consisting of nodes $n \in N$ and directed links $a \in A$. Let $P \subset N \times N$ be the set of OD pairs. Assume that OD demands q_p^k between the OD pairs $p \in P$ for each user class $k \in K$ are given and that to each link a are associated continuously differentiable cost functions $c_a^k : \mathbb{R}_+^K \rightarrow \mathbb{R}_+$, that give the cost of traversing link a for a user in class k dependent on the class specific volumes on the link. For the time being, c_a^k is a general function, but it will be endowed with a special structure in the next section.

Let R_p be the set of routes connecting OD pair p and $R = \cup_{p \in P} R_p$, the set of all routes. Let $H = \{h \in \mathbb{R}_+^{R \times K} : \sum_{r \in R_p} h_r^k = q_p^k, \forall p \in P, k \in K\}$ denote the set of feasible route flow vectors, and let $F = \{f \in \mathbb{R}_+^{A \times K} : f_a^k = \sum_{r \in R} \delta_{ra} h_r^k, \forall a \in A, k \in K, h \in H\}$, be the set of feasible link flows, where δ_{ra} is 1 if route r traverses link a , 0 otherwise.

Definition 1. (extended Wardrop) $\hat{h} \in H$ is a multi-class equilibrium flow if for any OD-pair p and user class k , the class specific costs of routes actually used (i.e. having $\hat{h}_r^k > 0$), are equal and not larger than those of any unused routes.

Similarly to the single user class case (see [6], Thm 3.14), the equilibrium condition can be written as a variational inequality in the set of link flows

$$\langle c(\hat{f}), f - \hat{f} \rangle \geq 0 \quad \forall f \in F \quad (1)$$

where $c = (c_a^k)_{a \in A, k \in K}$, (see [4]).

3 Equilibria under Social Marginal Cost Pricing

For space reasons this section is quite terse; for more details, see [3]. In the remainder of the paper, it is assumed that the drivers' perceived link travel cost consists of two components: the toll p_a and the travel time t_a , which is a twice differentiable function of the total link volume $f_a^{\text{tot}} = \sum_{k \in K} f_a^k$. Using class specific *time values* $v_k > 0$, the perceived travel cost can be expressed

either in time or in monetary terms. Thus we define *generalized time* \bar{t}_a^k and *generalized cost* \bar{c}_a^k of link a for class k respectively as

$$\bar{t}_a^k(f_a) = t_a(f_a^{tot}) + p_a/v_k, \quad (2a)$$

$$\bar{c}_a^k(f_a) = v_k t_a(f_a^{tot}) + p_a. \quad (2b)$$

Under SMC pricing, the users have to pay for the delays they incur on other users and one is interested in the traffic volumes that are established in the network and the corresponding toll values. The link toll then is the sum of all delay values for the users of the link caused by a marginal user, i.e.

$$p_a = p_a(f) = t'_a(f_a^{tot}) \sum_k v_k f_a^k. \quad (3)$$

The SMC equilibrium problem can now be formulated as the VIP (1) with link cost functions defined by inserting (3) into (2a) or (2b). Using the costs $c_a^k = \bar{c}_a^k$ (but not \bar{t}_a^k ! See [3], [4]), the VIP (1) corresponds to a necessary condition for a local minimum over F of the *total perceived value of travel time*,

$$V(f) = \sum_a t_a(f_a^{tot}) \sum_k v_k f_a^k, \quad (4)$$

since it has the property that $\nabla V(f) = \bar{c}(f)$, ([3], [4]).

In general, $V(f)$ is not convex and multiple SMC equilibria corresponding to different values of V can exist ([3], [4]).

4 The SMC Equilibrium Problem

According to the previous section equilibria (in the link flow space) can be determined by minimizing $V(f)$ over F . Introducing $w_a = \sum_k f_a^k v^k$ this problem can be stated

$$\begin{aligned} \min_{f, w, h} \quad & V(f) = \sum_a t_a(f_a^{tot}) \cdot w_a \\ \text{s.t.} \quad & \begin{cases} f_a^{tot} = \sum_k f_a^k \\ f_a^k = \sum_r \delta_{ar} h_r^k \\ w_a = \sum_k v^k f_a^k \\ \sum_{r \in R_p} h_r^k = q_p^k \\ h_r^k \geq 0 \end{cases} \end{aligned}$$

Using $C_a(f, w) =_{df} t_a(f) \cdot w$, we have $V(f) = \sum_a C_a(f_a^{tot}, w_a)$. We will analyze C_a , dropping the subscripts a for convenience. Thus let

$$\begin{aligned} C(f, w) &= t(f) \cdot w \text{ with gradient } \nabla C = (t'(f)w, t(f)) \text{ and hessian} \\ \nabla^2 C &= \nabla(\nabla C^T) = \begin{pmatrix} t''(f)w & t'(f) \\ t'(f) & 0 \end{pmatrix} \end{aligned}$$

Assuming $t''(f)w > 0$ and $t'(f) > 0$ (which is natural) this implies that $\nabla^2 C$ has one negative and one positive eigenvalue (as is easily checked). Hence C is generically nonconvex.

Let \underline{v} and \bar{v} be respectively the smallest and largest values of $v_k, k \in K$. Then, with f denoting f_a^{tot} , we have

$$w = w_a = \sum v_k f_a^k \leq \sum \bar{v} f_a^k = \bar{v} \cdot f_a^{tot} = \bar{v} f \text{ and symmetrically } w \geq \underline{v} f.$$

Thus, one may think of C as defined on the cone $K = \{(f, w) | \underline{v} f \leq w \leq \bar{v} f, f \geq 0\}$ bounded by two rays, $\bar{L} = \{(f, w) | w = \bar{v} f, f \geq 0\}$ and $\underline{L} = \{(f, w) | w = \underline{v} f, f \geq 0\}$.

5 Convexification of C

We will construct the convex hull, $convC$, of C on K . $convC$ is the largest convex minorant to C ([7], Ch. I)

We will perform this convexification for the commonly used travel times $t(f)$ of BPR type, i.e. $t(f) = t_0 + b \cdot f^c$, where t_0 is the free flow time, and b and c are appropriate positive constants. The convexification can be performed in a similar way for other functional forms. With such travel times, C can be written

$$C(f, w) = t_0 w + b f^c w, \text{ or, introducing } \bar{C}(f, w) =_{df} f^c w,$$

$$C(f, w) = t_0 w + b \cdot \bar{C}(f, w).$$

Since $t_0 w$ is an affine function, $convC(f, w) = t_0 w + b \cdot conv\bar{C}(f, w)$. Hence, it is enough to find $conv\bar{C}$. \bar{C} has the hessian

$$\bar{H} =_{df} \nabla^2 \bar{C} = \begin{pmatrix} c(c-1)f^{c-2}w & c f^{c-1} \\ c f^{c-1} & 0 \end{pmatrix} = c f^{c-2} \begin{pmatrix} (c-1)w & f \\ f & 0 \end{pmatrix},$$

which also has one positive and one negative eigenvalue, implying that \bar{C} locally is concave in some direction at each nonzero point $(f, w) \in K$. For a given point (\bar{f}, \bar{w}) in K we have ([7] Cor.17.1.5)

$$conv\bar{C}(\bar{f}, \bar{w}) = \inf_{\lambda_i, f_i, w_i} \{ \sum \lambda_i \bar{C}(f_i, w_i) | \sum \lambda_i = 1, \lambda_i \geq 0, (f_i, w_i) \in K, \sum \lambda_i f_i = \bar{f}, \sum \lambda_i w_i = \bar{w} \}$$

However, $\bar{C}(f, w) = f^c \cdot w \geq f^{c+1} \cdot \underline{v}$, implying that $\bar{C}(f, w) \rightarrow \infty$ when $f \rightarrow \infty$ in K . Hence the infimum above can be exchanged for a minimum:

$$\begin{aligned} conv\bar{C}(f, w) &= \min \{ \sum \lambda_i \bar{C}(f_i, w_i) | \sum \lambda_i = 1, \\ &\lambda \geq 0, (f_i, w_i) \in K, \sum \lambda_i f_i = \bar{f}, \sum \lambda_i w_i = \bar{w} \} \\ &= \sum \bar{\lambda}_i \bar{C}(\bar{f}_i, \bar{w}_i) \text{ for appropriate } \bar{\lambda}_i \geq 0, \text{ summing to 1 and } (\bar{f}_i, \bar{w}_i) \in K \end{aligned}$$

By the local (directional) concavity of \bar{C} , no point (\bar{f}_i, \bar{w}_i) can belong to $intK$, the interior of K . If this were the case, we could exchange (\bar{f}_i, \bar{w}_i) for a pair $(\bar{f}_i, \bar{w}_i) \pm (d_f, d_w)$, with weights $\bar{\lambda}_i/2$, where (d_f, d_w) is a direction of concavity at (\bar{f}_i, \bar{w}_i) . This would give a lower value to $\sum \lambda_i \bar{C}(f_i, w_i)$ contradicting that it is minimal.

Hence all (\bar{f}_i, \bar{w}_i) belong to \underline{L} or \bar{L} . But along \underline{L} and \bar{L} , \bar{C} is strictly convex, whence the minimum is achieved for a single point each on \underline{L} and \bar{L} (Fig 1).

We have proved

Proposition 1 For $(\bar{f}, \bar{w}) \in intK$,

$$conv\bar{C}(\bar{f}, \bar{w}) = \inf_{\lambda, f_1, f_2} \{ (1 - \lambda) \bar{C}(f_1, \underline{v} f_1) + \lambda \bar{C}(f_2, \bar{v} f_2) |$$

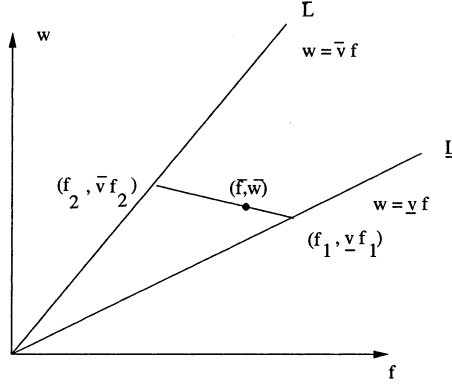


Fig. 1. Construction of $\text{conv}V$

$$f_i \geq 0, \lambda \in (0, 1), \bar{f} = (1 - \lambda)f_1 + \lambda f_2, \bar{w} = (1 - \lambda)\underline{v}f_1 + \lambda \bar{v}f_2\}$$

To determine the exact form of $\text{conv}\bar{C}$, we will construct it in the following way. We take an arbitrary point $(f, \underline{v}f)$ on \underline{L} . Through the corresponding point $P = (f, \underline{v}f, \bar{C}(f, \underline{v}f))$ on the graph of \bar{C} on \underline{L} we construct a tangent plane to that graph. We then rotate the tangent plane until it touches the graph of \bar{C} on \bar{L} at some point Q . The line segment from P to Q will then lie in the graph of $\text{conv}\bar{C}$.

The details of this construction is given [5]. We give the result.

Proposition 2 $\text{conv}C(f, w) = t_0 w + b\bar{h}((f, w)g)$ where

$\bar{h}(x) = \bar{v} \cdot (x/(\bar{v} - \underline{v}))^{c+1}$ and

$g = (\underline{v}(\rho^c - 1), 1 - 1/\rho)^T$, with $\rho = (\bar{v}/\underline{v})^{1/(c+1)}$

Corollary

$$\nabla \text{conv}C = (0, t_0) + b\bar{h}'((f, w)g)g^T$$

$$\nabla^2 \text{conv}C = b\bar{h}''((f, w)g)gg^T.$$

Proof By direct differentiation. \square

Letting $f_a = (f_a^k)_{k \in K}$ be a column vector, we have $f_a^{\text{tot}} = \sum_k f_a^k = e^T f_a$, with $e^T = (1, 1, \dots, 1)$, and $w = \sum v_k f_a^k = v^T f_a$. Thus, with $\bar{e} = (e, v)$,

$$f_a^T \bar{e} = f_a^T (e, v) = (f_a^T e, f_a^T v) = (e^T f_a, v^T f_a) = (f_a^{\text{tot}}, w_a) \quad (2)$$

Let $V_a(f_a) = t_a(f_a^{\text{tot}}) \cdot \sum v_k f_a^k$ be a generic term in $V(f)$. Then $\text{conv}V(f) = \sum \text{conv}V_a(f_a)$ since the terms have disjoint sets of variables. Further, $V_a(f_a) = C_a((f_a^{\text{tot}}, w_a)^T) = C_a(\bar{e}^T f_a)$

Hence, $\text{conv}V_a(f) = \text{conv}C_a(\bar{e}^T f_a)$, whence $\text{conv}V(f) = \sum \text{conv}C_a(\bar{e}^T f_a)$

Using the above relations (2), we can now evaluate the derivatives of $\text{conv}V_a$ w.r.t f_a :

$\text{conv}V_a(f_a) = \text{conv}C_a(\bar{e}^T f_a) = (0, t_a^0)\bar{e}^T f_a + b\bar{h}(g^T \bar{e}^T f_a) = \bar{t}_a^0 f_a + b\bar{h}(\bar{g}^T f_a)$, where $\bar{t}_a^0 = (0, t_a^0)\bar{e}^T$ and $\bar{g}^T = g^T \bar{e}^T$

Thus

$$\nabla V_a = \bar{t}_a^0 + b\bar{h}'(\bar{g}^T f_a)\bar{g}^T \text{ and}$$

$$\nabla^2 V_a = b\bar{h}''(\bar{g}^T f_a)\bar{g}\bar{g}^T.$$

6 Some Experimental Results

We have applied our results to two test cases, the classical Sioux Falls network (24 nodes, 76 links, 528 OD-pairs) and a study of Stockholm (1250 origins, 19539 links). The Stockholm case has three user classes (work, business, others) with time values (.98, 3.30, .19) and global fractions (.754, .036, .210). In Sioux Falls we used these same time values and applied these same fractions for all OD-demands.

The use of convexification may be threefold

- a. getting a valid lower bound,
- b. determining a promising starting point, and avoiding getting trapped in a bad local optimum, by initially minimizing $\text{conv}V$, and finally
- c. (related to the previous point) possibly achieving faster initial convergence by minimizing $\text{conv}V$ instead of V .

6.1 Sioux Falls Network

Previous runs ([4]) with Frank-Wolfe on V (in a Matlab environment) indicate that V has only one local, and hence global, optimum. Hence usage point b is not relevant here.

The lower bounds for V and $\text{conv}V$, achieved by long runs are 63.810 and 46.719 respectively. Hence, the convexity gap is discouragingly large.

In Figure 2 we display on the one hand (2b) how the relative errors develop with the iterations when minimizing V and $\text{conv}V$ respectively, and, on the other hand (2a) the errors for V when minimizing V and $\text{conv}V$ respectively. It is interesting to note that for iterations 15-70 we achieve better points for V , when minimizing $\text{conv}V$. In the early iterations, however, minimization of $\text{conv}V$ is not beneficial.

The relative errors are determined as $(V(f^{(i)}) - LBD)/LBD$ (where LBD is the linearization bound at iteration $i = 10000$), and correspondingly for $\text{conv}V$.

6.2 Stockholm

Stockholm is a rather large case, and the iterations are heavy. We have implemented the Frank-Wolfe method for minimizing V and $\text{conv}V$ using macros in EMME/2.

The case concerns the potential addition of a major eastern bridge/tunnel system connecting the north and south parts of the city (otherwise separated by water). The comparison is made with and without SMC tolls, giving in total 4 cases. For Stockholm the convexity gaps are much smaller (on the order of 3 %) as can be seen in Table 1.

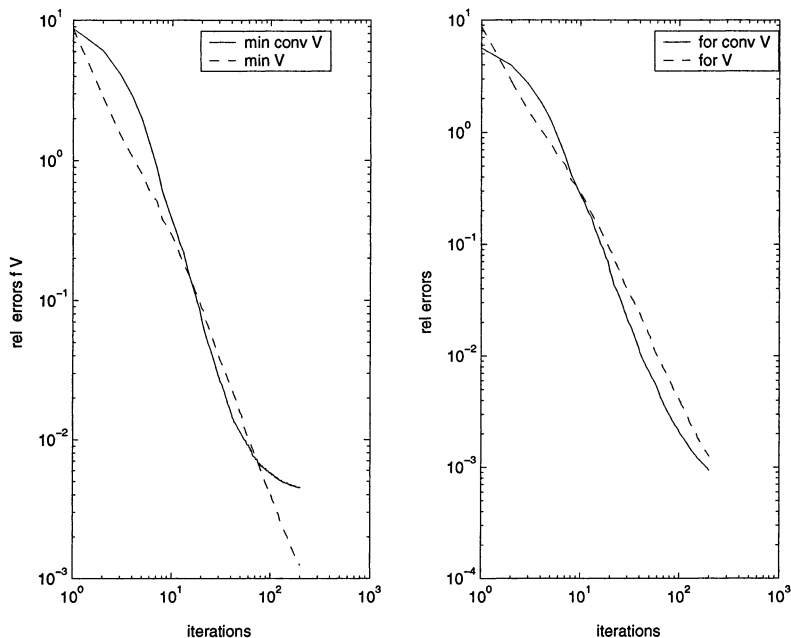


Fig. 2. Iteration histories of SMC calculations **a.**(left) rel errors for V , minimizing V or $\text{conv}V$, and **b.**(right) rel errors for V and $\text{conv}V$

	Without bridge/tunnel	With bridge/tunnel
Without tolls	5.176	5.078
With tolls	5.119	5.035
Lower bound	4.950	4.890

Table 1 Results for Stockholm (societal cost in MSEK/h at busy hour).

Concerning uses b and c , minimization of $\text{conv}V$ instead of V seems worthwhile, leading to smaller relative errors in the beginning (Fig. 3). So for the production runs, we start with minimizing $\text{conv}V$ for the first 10 iterations, then switching to V . The Stockholm case has been too heavy to allow us to study the existence of multiple local optima (usage b)

References

1. Beckmann, M, McGuire, C, and Winsten, C. (1956) *Studies in the economics of transportation*. Yale University Press, New Haven.
2. Dafermos, S.(1973) Toll patterns for multiclass-user transportation networks. *Transportation Sci.* **7**, 211–223.

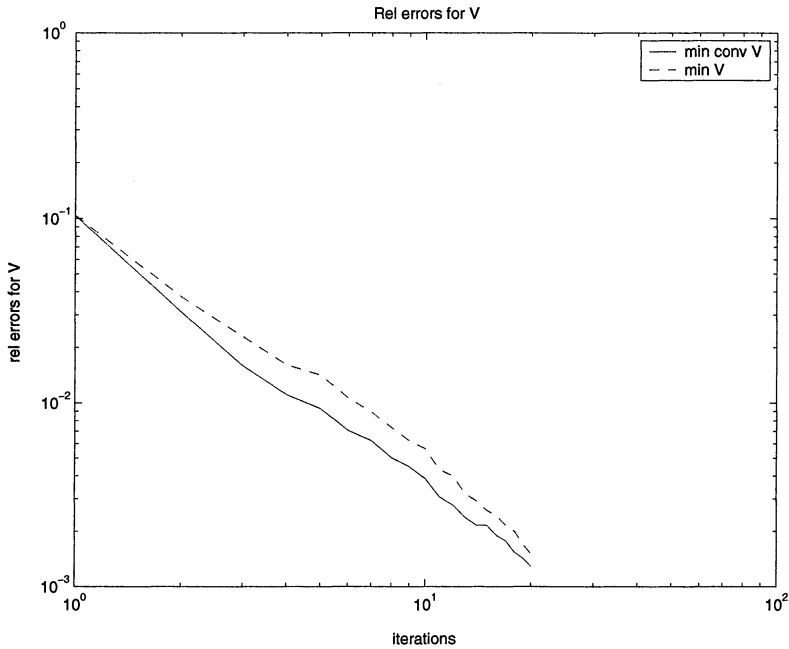


Fig. 3. Rel. errors for V for Stockholm, minimizing solid: $\text{conv}V$, dashed: V

3. Engelson, L., Lindberg, P.O. (2002) Congestion Pricing of Road Network Users with Different Time Values, Technical Report LiTH-MAT-R-2002-10, revised version LiTH-MAT-R-2003-12, Linköping University, forthcoming.
4. Engelson, L., Lindberg, P.O., and Daneva, M. (2003) Congestion Pricing of Road Network Users with Different Time Values, *Operations Research Proceedings 2003*, U. Leopold-Wildburger, F. Rendl and G. Wäscher (Eds), Springer, 174–179.
5. Lindberg, P.O. and Engelson, L (2003) Convexification of the Traffic Equilibrium Problem with Social Marginal Cost Tolls, Technical Report LiTH-MAT-R-2003-13, Linköping University, submitted.
6. Patriksson, M. (1994) *The Traffic Assignment Problem – Models and Methods*. VSP, Utrecht, The Netherlands.
7. Rockafellar, R.T. (1970) *Convex Analysis*. Princeton University Press, Princeton, N.J.
8. Van Vliet, D. (1986) Equilibrium Traffic Assignment with Multiple User Classes. *Proceedings PTRC 14th Summer Annual Meeting*, 111–122.

Vermittlung von Fahrgemeinschaften betrachtet als Vehicle Routing Problem*

Gerriet Reents

Carl von Ossietzky Universität Oldenburg, Department für Informatik
26111 Oldenburg
e-mail: gerriet.reents@informatik.uni-oldenburg.de

Zusammenfassung Während für Fahrgemeinschaften auf Langstrecken häufig einfache Vermittlungsverfahren ausreichen, kann und muss die Vermittlung von Fahrgemeinschaften auf Kurzstrecken als Online-Optimierungsproblem dargestellt werden. Ausgehend von einer Einordnung des als ganzzahliges lineares Programm formulierten Problems in ein Klassifikationsschema für Tourenplanungsprobleme werden hier heuristische und exakte Lösungsalgorithmen vorgestellt, die einerseits die Anwendbarkeit unter praktischen Online-Bedingungen ermöglichen, aber – mangels geeigneter Standard-Benchmarks – gleichzeitig auch eine Bewertung der Qualität der Lösungen erlauben.

1 Einleitung

Eine Vielzahl realer logistischer Problemstellungen lassen sich als Tourenplanungsprobleme (Vehicle Routing Problems, VRP) [3] modellieren. Auch die Vermittlung von Fahrgemeinschaften im Berufsverkehr kann als ein spezielles VRP angesehen werden. Während in der Praxis für die Vermittlung von Fahrgemeinschaften auf Langstrecken häufig Ansätze, die auf einfachen Datenbankabfragen basieren, erfolgreich sind, gilt dies für Vermittlungen im Berufsverkehr nicht, da hier deutlich höhere Anforderungen an die Einhaltung räumlicher und zeitlicher Abhängigkeiten gestellt werden. In diesem Beitrag wird der im Fahrgemeinschaftsvermittlungssystem der Universität Oldenburg¹ gewählte Ansatz in Hinblick auf die Modellierung des Vermittlungsproblems als VRP und die verwendeten Algorithmen beschrieben. Einen Überblick über das Gesamtsystem liefert [8]. Während der Entwicklung dieses Systems sind ähnliche Ansätze wie etwa M21 [5] und PTV RideShare [6] entstanden. Auch hier wird die Vermittlung von Fahrgemeinschaften als Tourenplanungsproblem aufgefasst und mit entsprechenden Algorithmen gelöst.

2 Problemeigenschaften

Tourenplanungsprobleme treten in einer Vielzahl von Variationen auf. Daher soll das hier behandelte Problem mit Hilfe eines Klassifikationssystems nach

* gefördert durch die Deutsche Bundesstiftung Umwelt (DBU)

¹ siehe www.fgm.uni-oldenburg.de. Geplante Aufnahme des Betriebs im Oktober 2003.

Domschke [3] eingeordnet werden. Bei der Vermittlung von Fahrgemeinschaften werden Anfragen als *Angebote von* und *Gesuche nach* Fahrgemeinschaften betrachtet. Beide Arten von Anfragen haben gemeinsame Eigenschaften:

- jeweils einen Startort sowie einen Zielort;
- neben dem eigentlichen Startort können weitere alternative Aufnahmepunkte (etwa Parkplätze an Autobahnauffahrten) existieren;
- Zeitfenster sowohl für den Start- als auch für den Zielort, die nicht verletzt werden dürfen;
- Assoziationen zwischen Hin- und Rückfahrt wegen alternativer Aufnahmepunkte.

Ein Angebot stellt jeweils ein Fahrzeug mitsamt Depot mit folgenden Eigenschaften zur Verfügung:

- fahrzeugspezifische Kapazität (i.d.R. zwischen 2 und 5 Personen);
- Routen dürfen keine längeren Umwege enthalten, als vom Fahrer toleriert;
- Mindestmitfahrzeiten;
- Via-Punkte, die auf der Route liegen sollen.

Außerdem kann ein Fahrer angeben, ob er ggf. auch an einer anderen Fahrgemeinschaft als Mitfahrer teilnehmen möchte. Neben diesen auf Fahrzeuge und Routen bezogenen Eigenschaften können sich Nutzer des Systems gegenseitig bewerten und bestimmen, dass sie mit bestimmten anderen Nutzern keine Fahrgemeinschaften bilden möchten. Abbildung 1 enthält ein Beispiel für die Tourenstruktur.

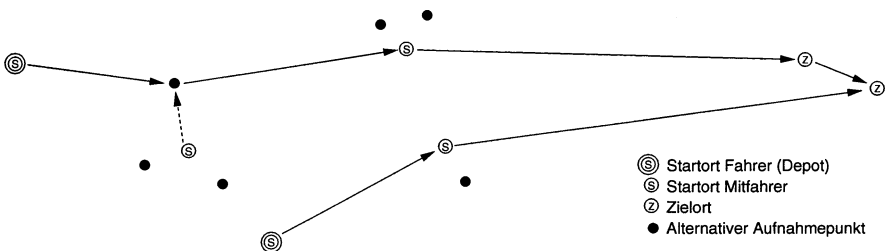


Abbildung 1. Beispielszenario

Gesucht wird eine Zusammenstellung von Fahrgemeinschaften, die zum einen die insgesamt zurückgelegte Wegstrecke minimiert und zum anderen für in der Summe möglichst positive gegenseitige Bewertungen der Fahrgemeinschaftsteilnehmer sorgt. Daneben sollen möglichst viele Gesuche vermittelt werden. Diese Problemeigenschaften lassen sich auf ein formales Optimierungsproblem, das als gemischt ganzzahliges lineares Programm (Mixed Integer Programm, MIP) [7] vorliegt, abbilden.

In dem von Domschke [3] vorgeschlagenen Klassifizierungssystem lässt sich das Fahrgemeinschaftsvermittlungsproblem wie folgt charakterisieren:

$$[p, \text{task}, p/d, \text{tw}, \text{sel} | \text{cap}_i, \text{dur}_i | \text{dir}, K/F | L]$$

Hierbei beschreibt das 4-Tupel $[\alpha | \beta | \delta | \gamma]$ die verschiedenen Problemcharakteristiken. α steht für die Depot- und Kundencharakteristik, β für die Fahrzeugcharakteristik, δ für die Problem- oder Zusatzcharakteristik und γ für die Zielsetzungen. Im einzelnen bedeutet dabei:

Depot- und Kundencharakteristik

- p Es gibt p Depots, nämlich die Startorte der Fahrgemeinschaftsanbieter. Zu diesen Depots muss nicht zurückgekehrt werden.
- task Die zu befördernden Güter – hier Mitfahrer – werden an einem bestimmten Ort aufgenommen und an einem anderen wieder abgesetzt.
- p/d Keine reine Belieferung oder Abholung
- tw Harte Zeitfenster für Abholung *und* Belieferung
- sel Nicht alle Kunden müssen bedient werden.

Fahrzeugcharakteristik

- cap_i Fahrzeuge mit unterschiedlichen Kapazitäten. In der Regel zwischen 2 und 5 Plätzen.
- dur_i Maximale Tourdauer pro Fahrzeug beschränkt durch Umwegeranz

Problem- oder Zusatzcharakteristik

- dir gerichteter Graph
- K/F Nicht jeder Kunde kann durch jedes Fahrzeug bedient werden, da Kunden andere Kunden ausgeschlossen haben könnten.

Zielsetzungen

- L Minimierung der zurückgelegten Entfernungen

Einige Problemeigenschaften lassen sich durch obiges System nicht ausdrücken. So ergibt sich als zusätzliche Fahrzeugcharakteristik, dass Fahrzeuge nicht zu ihren Depot zurückkehren und nicht warten können, bis ein Zeitfenster eingehalten wird. Die Existenz alternativer Aufnahmepunkte führt einfache Umladevorgänge ein. Durch die möglichen Bewertungen ergibt sich außerdem eine weitere Zielsetzung.

Da das Fahrgemeinschaftsvermittlungsproblem innerhalb eines Online-Systems gelöst werden muss, ergeben sich weitere Rahmenbedingungen für die verwendeten Algorithmen. Zum einen sollten Ergebnisse innerhalb gewisser Zeitschranken sicher vorliegen – was keinen Einfluss auf die formale Problemdefinition hat. Zum anderen liegen zu lösenden Probleminstanzen nie vollständig vor, da sich fortlaufend Änderungen durch Benutzerinteraktionen ergeben. Da auf der Basis der zu einem bestimmten Zeitpunkt vorliegenden

Ergebnisse jedoch auch Vermittlungsentscheidungen getroffen und nach außen kommuniziert werden müssen, die nicht wieder zurückgenommen werden dürfen, ist es nötig, Teile der Lösung fixieren zu können, was in der Problemdefinition berücksichtigt werden muss.

3 Lösungsalgorithmen

Die verschiedenen Lösungsalgorithmen, die bisher entstanden sind, beachten jeweils die oben beschriebene Problemdefinition, was ihre Leistungsfähigkeit vergleichbar macht. Da die vorgegebenen Schranken für die Laufzeit der im Vermittlungssystem verwendeten Algorithmen aufgrund der Problemkomplexität nur durch heuristische Ansätze sicher eingehalten werden können, zur Einschätzung der Qualität der Heuristiken aber auch Kenntnis über optimale Problemlösungen nötig war, wurden sowohl heuristische als auch exakte Verfahren entwickelt. Diese werden in den folgenden beiden Abschnitten beschrieben.

3.1 Heuristische Verfahren

Es existieren zwei Heuristiken: eine verhältnismäßig einfache Greedy-Heuristik sowie ein Tabu-Suche Verfahren. Die *Greedy-Heuristik* arbeitet in drei Schritten:

1. Priorisierung der Gesuche
2. Einfügen des wichtigsten Gesuchs in die am besten passende Fahrgemeinschaft
3. Nachoptimierung

Im ersten Schritt wird gemäß einiger statisch prüfbarer Kriterien, wie die grundsätzliche Kompatibilität der Zeitfenster oder wie mögliche Ausschlüsse anderer Teilnehmer, eine Reihenfolge ermittelt, in der versucht wird, die Gesuche passenden Fahrgemeinschaften zuzuordnen. Dies geschieht dann im zweiten Schritt. Für jedes Gesuch wird geprüft, ob und wie gut es in eine der bisher konstruierten Fahrgemeinschaften passt. Dazu wird der optimale Pfad über die Orte, die von der Fahrgemeinschaft angesteuert werden, durch Enumeration mittels einfachen Branch&Bounds errechnet. Diese Strategie ist hier möglich, da Fahrgemeinschaftstouren nur eine begrenzte Länge aufweisen. Selbst vollbesetzte Pkw ergeben selten Pfade mit mehr als 10 Orten. Die Fahrgemeinschaft, die sich gemäß Umweg und der Bewertung der Fahrgemeinschaftsteilnehmer untereinander am besten eignet, wird ausgewählt und das Gesuch dieser Fahrgemeinschaft fest zugeordnet. Diese Entscheidung wird gemäß der Greedy-Strategie im weiteren Verlauf nicht wieder verändert. Sind auf diese Weise alle Gesuche abgearbeitet, findet im dritten Schritt eine Nachoptimierung statt. Hier werden durch Austausch von je einem Mitfahrer zwischen zwei Fahrgemeinschaften im Sinne eines 2-opt-Verfahrens mögliche Verbesserungen ermittelt. Anzumerken ist, dass die Greedy-Heuristik die

Möglichkeit den Fahrer eines Angebots auch als Mitfahrer in eine andere Fahrgemeinschaft zu vermitteln nicht nutzt.

Die Strategie für einzelne Fahrgemeinschaften bei bekannter Zusammensetzung optimale Touren zu bestimmen, hat für den Betrieb in der Praxis einen wichtigen Vorteil: Die Mitglieder einer Fahrgemeinschaft werden nicht durch etwaige Artefakte der heuristischen Herangehensweise verstört. Die Entscheidungen des Systems, die die einzelnen Mitglieder überblicken können – nämlich die Zusammenstellung ihrer Fahrgemeinschaft –, sind immer optimal. Gründe für eine Abweichung vom Optimum liegen immer in der gewählten Zuordnung der Gesuche zu den Angeboten.

Eine zweite Heuristik, die nach dem Prinzip der *Tabu-Suche* [4] arbeitet, erweitert den Ansatz der Greedy-Heuristik. Nach einer initialen Phase, die den ersten beiden Schritten des oben beschriebenen Verfahrens gleicht, findet eine schrittweise Verbesserung der Zwischenlösung statt. Als mögliche Veränderungsschritte dienen

1. das Entfernen eines Mitglieds einer Fahrgemeinschaft und Einfügen in eine neue,
2. die Auflösung der Fahrgemeinschaft eines Fahrer, der bereit ist, in einer anderen Fahrgemeinschaft mitzufahren und
3. die Neugründung einer Fahrgemeinschaft, falls ein Fahrer bisher als Mitfahrer fungiert.

Der erste Schritt entspricht den einzelnen Veränderungen bei der 2-opt Nachoptimierung der Greedy-Heuristik. Er lässt sich durch doppelt Anwendung negieren. Die Schritte 2 und 3 führen zu einer Erweiterung gegenüber der Greedy-Heuristik. Sie lassen sich durch den jeweils anderen wieder zurücknehmen. Um den Aufwand, der zum Durchsuchen der gesamten Nachbarschaft einer Zwischenlösung nötig ist, zu verringern, wird eine Kombination der von Glover [4] vorgeschlagenen Strategien *Aspiration-Plus* und *Elite-Candidate-List* verwendet.

3.2 Exakte Verfahren

Da Heuristiken zwar gemäß der Problemdefinition gültige, aber nicht notwendigerweise optimale Ergebnisse liefern, wurde auch ein exaktes Verfahren entwickelt. So kann überprüft werden, ob die Heuristiken eine ausreichende Qualität besitzen. Dieses Verfahren basiert direkt auf der Problemformulierung als MIP, während diese für die Heuristiken eher als Spezialisierung gilt.

Als Lösungsansatz dient hier die für Tourenplanungsprobleme übliche Dekomposition in zwei Teilprobleme – dem Master- und dem Subproblem –, die durch einen Branch&Price-Algorithmus gekoppelt werden (siehe z.B. [2]). Das Master-Problem ist ein beinahe reines Set-Partitioning-Problem. Es gilt aus einer Menge von möglichen Fahrgemeinschaften, deren Struktur und Beitrag

zur Zielfunktion bekannt ist, diejenigen auszuwählen, die möglichst viele Kunden bedienen und gleichzeitig die Zielfunktion minimieren. Bedingt durch die nötige Kopplung bestimmter Hin- und Rückfahrten, die zueinander passende Eigenschaften haben, existieren hier über das Set-Partitioning-Problem hinausgehende Nebenbedingungen. Das Master-Problem wird durch einen Branch&Bound-Algorithmus mit LP-Relaxation als unterer Schrankenfunktion gelöst.

Das Sub- oder auch Pricing-Problem erbt annähernd alle Nebenbedingungen der originalen Problemdefinition. Es stellt ein Kürzeste-Wege-Problem mit Nebenbedingungen dar. Zur Lösung wird ein Algorithmus der Dynamischen Programmierung eingesetzt. Die Kopplung des Subproblems an das Masterproblem führt dazu, dass die dualen Kosten der Nebenbedingungen einer Lösung der LP-Relaxation mit den Kantengewichten im Subproblem verrechnet werden. Dadurch können negative Kantengewichte und ebenfalls negativ bewertete Zyklen im Graphen entstehen. Diese Eigenschaften erschweren die Lösung des Problems, da kein Label-Setting sondern ein Label-Correcting Algorithmus eingesetzt werden muss. Außerdem zeigt sich, dass sich gegenüber reinen Tourenplanungsproblemen wie dem klassischen VRP mit Zeitfenstern [2] kein starkes Dominanzkriterium für Knoten-Label finden lässt. Dort kann durch die möglichen Wartezeiten ein Label eine größere Spanne von Labeln dominieren. Beim hier betrachteten Problem ohne Wartemöglichkeit funktioniert dies nicht. Darüber hinaus wird durch die Menge der in den Knoten-Labeln gespeicherten Informationen (neben Ankunftszeit, aktueller Beladung z.B. auch die Zusammensetzung der Fahrgemeinschaft) die Definition einer wirksamen Dominanz erschwert. Derzeit wird neben diesem Multi-Label-Shortest-Path-Algorithmus ein weiterer Pricing-Algorithmus entwickelt, der als Branch&Cut-Algorithmus konzipiert ist. Hier werden die an Netzwerkflussprobleme angelehnten Nebenbedingungen sowie einige Klassen von problemspezifischen Nebenbedingungen statisch erzeugt. Alle Kurzzyklus-Eliminations-, sowie zeit- und kapazitätsabhängigen Nebenbedingungen werden durch Schnittebenenverfahren sukzessive hinzugefügt. Dabei werden Zeit und Kapazität nicht als Variable für jeden Knoten modelliert; die diese Variablen betreffenden Nebenbedingungen werden statt dessen über ungültige Pfade (siehe Ascheuer in [1]) realisiert. Pfade, die zeit- oder kapazitätsabhängige Nebenbedingungen verletzen, werden über entsprechende Schnittebenen verboten. Derzeit ist der Branch&Cut-Ansatz allerdings noch nicht leistungsfähig genug, um mit dem Multi-Label-Shortest-Path-Algorithmus konkurrieren zu können.

4 Ergebnisse

Die Leistungsfähigkeit der bisher entwickelten Algorithmen ist in Tabelle 1 dargestellt. Aufgrund der vielen Detaileigenschaften, die das Fahrgemeinschaftsvermittlungsproblem aufweist, sind umfassende Leistungscharakteri-

sierungen recht aufwendig. Die hier vorgestellten Ergebnisse sollen zeigen, dass typische Anwendungsszenarien, wie sie im Betrieb an der Universität Oldenburg vorkommen können, von den Algorithmen beherrscht werden. Dazu dienen vier künstlich erzeugte Szenarien. Die in ihnen enthaltenen Anfragen – also Gesuche und Angebote – sind zeitlich kompatibel, da sie innerhalb eines kurzen Tageszeitraums stattfinden, können aber über mehrere Tage gestreut sein. In realen Daten dürften durch unterschiedliche Vorlesungszeiten größere Streuungen auftreten. Die Szenarien sind daher „Härtefälle“. Szenario 1 entspricht der erwarteten punktuellen Maximalbelastung. In den Szenarien 2.1 bis 2.3 wird die Entwicklung gezeigt, wenn das System ein wachsende Anzahlen von Anfragen bearbeiten muss, die Anzahl zeitlich kompatibler Anfragen pro Tag aber konstant bleibt. Es zeigt sich das der exakte Algorithmus

Tabelle1. Leistung der Vermittlungsalgorithmen. Mit * markierter Eintrag ist Abschätzung von unten, mit + markierter Eintrag Lösung < 101% des Optimums. System: Linux, i586 Prozessor mit 733 MHz Taktfrequenz. GH=Greedy-Heuristik, GH+NO=GH mit Nachoptimierung, TS=Tabu-Suche, OPT=exaktes Verfahren

Verfahren	Szenario			Qualität		Laufzeit [s]
	Nr. Anfragen Tage			Kumulierte Wegstrecke	Unvermittelte Teilnehmer	
GH	1	613	1	22708	55	1204
GH+NO						
TS				24119	50	2145
OPT						
GH	2.1	240	4	7833	16	28
GH+NO				6817	16	340
TS				6864	8	28
OPT				6392	8	676
GH	2.2	466	9	17245	18	142
GH+NO				15841	18	2925
TS				16147	9	157
OPT				*13709	9	
GH	2.3	971	19	36068	31	454
GH+NO				32955	31	23984
TS				33494	20	304
OPT				+28888	20	

tatsächlich nicht in der Lage ist, praxisrelevante Probleme (Szenario 1) zu lösen. Es kann nur ein einfaches Problem (Szenario 2.1) vollständig gelöst werden. In anderen Szenarien (2.2 und 2.3) sind jedoch noch Abschätzungen möglich, die aufgrund der Informationen, die während des Lösungsprozesses gewonnenen wurden, getroffen werden konnten. Der Vergleich der exakten Ergebnisse mit denen der Heuristiken zeigt, dass eine durchaus akzeptable Qualität vorliegt. Erkennbar ist auch, dass sich die Heuristiken gegenüber auf der

Zeitachse wachsenden Problemgrößen gutmütig verhalten. Diese Dimension ist besonders wichtig, da sie vor Start des Praxisbetriebs schlecht abschätzbar ist. Andere Dimension, wie die Verteilung der Studierenden oder die maximale Häufung von zeitlich passenden Anfragen, lassen sich eher ermitteln. Bei der exakten Lösung von Szenario 2.1 fällt als Randergebnis folgende Beobachtung auf: Es werden 655 der insgesamt 1400 möglichen Fahrgemeinschaften durch das Pricing geliefert und im Master-Problem verwendet. Erfreulich ist, dass die Tabu-Suche immer optimal viele Gesuche vermitteln kann.

5 Ausblick

Es konnte gezeigt werden, dass die für das Fahrgemeinschaftsvermittlungssystem der Universität Oldenburg entwickelten Algorithmen die Problemgrößen, die erwartet werden, verarbeiten können und dabei eine zufriedenstellende Qualität erreichen, die durch weitere Verbesserungen noch erhöht werden soll. Diese anhand künstlich generierter Szenarien ermittelten Erfahrungen können in der nächsten Zeit mit realen Daten überprüft werden.

Literatur

1. ASCHEUER, NORBERT: *Hamiltonian Path Problems in the On-line Optimization of Flexible Manufacturing Systems*. Doktorarbeit, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1996.
2. DESROSIERS, JACQUES, FRANCOIS SOUMIS und MARTIN DESROCHERS: *Routing with Time Windows by Column Generation*. Networks, 14, 1984.
3. DOMSCKE: *Logistik: Rundreisen und Touren*. Oldenbourg, 1997.
4. GLOVER, FRED und MANUEL LAGUNA: *Tabu Search*. Kluwer Academic Publishers, 1997.
5. HOLZWARTH, JÜRGEN, ALFRED BIESINGER und TORSTEN FUNKE: *M21 - Einführung neuer telematikgestützter Mobilitätsdienstleistungen für den Berufsverkehr im Ballungsraum*. Strassenverkehrstechnik, 10:549–555, 2000.
6. PTV AG: *Gern unterwegs mit PendlernetzPro*. Pressemitteilung, November 2002.
7. REENTS, GERRIET: *Formale Definition eines Fahrgemeinschaftsvermittlungsproblems als MIP*. Internes Arbeitspapier, 2001.
8. REENTS, GERRIET: *Carpooling in Commuter Traffic aided by an Internet Communication Platform*. In: W. PILLMANN, K. TOCHTERMANN (Herausgeber): *Environmental Communication in the Information Society: 16th International Symposium Informatics for Environmental Protection*, 2, Seiten 494–501. ISEP, 2002.

Single Machine Scheduling with Precedence Constraints and SLK Due Date Assignment

Gordon, V. ^a, Proth, J.-M. ^b, Strusevich, V. ^c

^a United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganov str. 6, 220012 Minsk, Belarus, Email: gordon@newman.bas-net.by

^b SAGEP Project, INRIA-Lorraine, 57012 Ile du Saulcy, Metz, France, Email: Jean-Marie.Prothgordon@loria.fr

^c School of Computing and Mathematical Sciences, University of Greenwich, London, UK, Email: V.Strusevich@greenwich.ac.uk

Abstract. A single machine due date assignment and scheduling problem of minimizing holding costs with no tardy jobs is considered under series-parallel and somewhat wider class of precedence constraints.

1. Introduction

For a single machine due date assignment and scheduling problem under consideration, due dates are determined by increasing the processing times of jobs by a common positive slack. The objective is to explore the trade-off between the size of the slack and the arising holding costs for the early jobs, and to find an optimal schedule that minimizes holding costs provided that there are no tardy jobs and precedence constraints are respected. The factors that may affect the complexity of scheduling problem include the structure of precedence constraints and the objective function involved. We consider the case when holding costs are represented by total weighted earliness or total weighted exponential earliness functions, and precedence constraints are defined by a class of graphs that is studied in [1, 9, 10, 12, 13, 15] and includes series-parallel graphs [2, 17].

Let the jobs of set $N = \{1, K, n\}$ have to be processed with no preemption on a single machine, and the processing of job $j \in N$ take p_j time units. The jobs are simultaneously available at time zero, and for each job j a weight w_j is given that indicates its relative importance. The machine can handle only one job at a time, and is permanently available from time zero. The machine is allowed to be idle if required. Each job j has the *due date* d_j by which it is desirable to complete its processing. Let C_j be the completion time of job j in a certain

schedule. Job j is *tardy* if $C_j > d_j$, and job is *early* if $C_j < d_j$ with $E_j = d_j - C_j$ being its *earliness*. A schedule is feasible if there are no tardy jobs and the job sequence respects given precedence constraints. We assume that the due dates are determined according to the so-called SLK rule [3], i. e., for each job the due date is obtained by adding a positive slack q to the processing time: $d_j = p_j + q$ for all $j \in N$. We look for the value of q which minimizes the objective function $\varphi(F, q)$ over the set of feasible schedules. Here $F = F(E_1, E_2, K, E_n)$ is an earliness penalty function and $\varphi(F, q)$ is an arbitrary non-decreasing function in both arguments. Extending standard scheduling notation [8], we refer to our problem as $1|prec, C_j \leq d_j = p_j + q|\varphi(F, q)$, where “*prec*” indicates the presence of arbitrary precedence constraints. In this paper we focus on the functions $F = \sum_{j=1}^n w_j E_j$ (total weighted earliness) and $F = \sum_{j=1}^n w_j \exp(\gamma E_j)$, where $\gamma \neq 0$ (total weighted exponential earliness). Once q is chosen or fixed, the corresponding scheduling problem $1|prec, C_j \leq d_j = p_j + q|F$ without due date assignment is to find a feasible schedule with the minimum value of function F . A general scheme for the due date assignment problems with arbitrary precedence constraints and arbitrary functions is presented in [6], and it is shown how to implement this scheme to obtain $O(n^2 \log n)$ – time algorithms for series-parallel precedence constraints provided that F is either the sum of linear functions or the sum of exponential functions.

In this paper, we show that our due date assignment and scheduling problem has a polynomial-time solution if the precedence constraints are represented by a more general class of graphs than series-parallel, namely, by the graphs which can be decomposed in such a way that the size (or width) of the “building blocks” or outer factors is limited.

The functions $\sum_{j=1}^n w_j E_j$ and $\sum_{j=1}^n w_j \exp(\gamma E_j)$ are closely related to the class of priority-generating functions which was first introduced in [4]. Systematic exposition of well-solved scheduling problems with priority-generating objective functions can be found in [16]. The priority-generating functions admit of the string interchange relation [8] and are studied also by Monma and Sidney [11] based on decomposition approach of Sidney [14]. Polynomial-time scheduling algorithms for this kind of objective functions and series-parallel precedence constraints are considered in [5, 7, 11]. In [10, 12, 15] some of these results have been generalized to a wider class of precedence constraints using an approach to modular decomposition of sequencing problems [1, 9, 13].

2. Precedence Constraints, Graphs and Posets

Formally, precedence constraints among the jobs are defined by a binary relation \rightarrow . We write $i \rightarrow j$ and say that job i *precedes* j if in any feasible schedule job i must be completed before j starts. Binary relation \rightarrow is a *strong order* relation, that is both asymmetric ($i \rightarrow j$ implies that not $j \rightarrow i$) and transitive ($i \rightarrow j$ and $j \rightarrow k$ implies $i \rightarrow k$). We write $i \sim j$ if jobs are *independent*, i.e. neither $i \rightarrow j$ nor $j \rightarrow i$. Precedence constraints are usually given by a directed circuit-free graph G in which the set of vertices is identical with the set of jobs and there is a path from vertex i to vertex j if and only if job i precedes job j . Recall that a graph is *circuit-free* (or *acyclic*) if it contains no cycles, i.e., contains no paths with the same initial and terminal vertices. Moreover, any directed acyclic graph (*dag*) induces a partial order on its vertices: $i \rightarrow j$ if and only if there is a path from vertex i to vertex j in G . The elements of a *partially ordered set* (*poset*) $P = (N, R)$ are given by the job set N , and ordering relations R are defined by $\langle i, j \rangle \in R$ if and only if $i \rightarrow j$. So, in our case, the notions of “precedence constraints”, “dag” and “poset” are interchangeable. For a dag, let (i, j) denote an arc that goes from vertex i to vertex j . The *transitive closure* of a dag G is a dag G_T such that G_T contains an arc (i, j) if and only if $i \neq j$ and there is a path from i to j in G .

Given a poset $P = (N, R)$, a subset $N' \subseteq N$ is a (*job*) *module* of P if for every job $k \in N \setminus N'$ one of the following holds: (a) $k \rightarrow i$ for all $i \in N'$, or (b) $i \rightarrow k$ for all $i \in N'$, or (c) $i \sim k$ for all $i \in N'$. Let $G = (N, U)$ be a dag corresponding to poset $P = (N, R)$. Replacing all arcs of the transitive closure G_T by undirected edges, we obtain the (undirected) graph $\tilde{G} = (N, E)$. We may assume that $\tilde{G} = (N, E)$ is given to us in the form of the adjacency matrix. A module N' is a set of vertices that is indistinguishable in graph \tilde{G} by the vertices outside N' ; that is, in graph \tilde{G} any vertex in $N \setminus N'$ is either adjacent to all vertices of N' , or is adjacent to no vertex in N' . There are three distinct types of modules: parallel, series, and neighborhood. To introduce these types of modules, we first introduce the notions of a complement graph and a complement-connected graph. The *complement* graph of $\tilde{G} = (N, E)$ is the graph (N, E') , where $(u, v) \in E'$ if and only if $(u, v) \notin E$. A graph is *complement-connected* if its complement graph is connected. *Parallel* modules are characterized by the property that the subgraph induced by the vertices of the module is not connected. A module is a *series* module if the subgraph induced by the vertices of the module

is not complement-connected. In a *neighborhood* module, the subgraph induced by the vertices of the module is both connected and complement-connected.

For a poset $P=(N,R)$, a subset $S \subseteq N$ is *initial* if for each $i \in S$ all predecessors of i are also in S . For an initial set S , define the set $I(S)=\{j: j \in S, j \text{ has no successors in } S\}$. The maximum size of any set $I(S)$ is called the *width* of P . If the poset is decomposed into modules, the size of any module does not exceed the width of the original poset. Provided that the objective function possesses certain properties, it can be minimized over a poset of a fixed width in polynomial time.

3. Scheduling by Modular Decomposition

When graph is divided into modules, the decomposition process is called *modular decomposition* [13]. At any stage of the process, the current subgraph being decomposed will be a module of the original graph. Each of these subgraphs is decomposed recursively. This process continues until all the subgraphs being decomposed contain only a single vertex. The decomposition procedure decomposes the graph uniquely. Parallel modules are decomposed into their connected components. Series modules are decomposed into their complement-connected components. Neighborhood modules are decomposed into maximal submodules, where a module M' is a *maximal submodule* of a neighborhood module M if M' is contained in M , and no proper submodule of M contains M' . Every vertex of M is contained in a unique maximal submodule.

To describe modular decomposition formally we shall use the following notation of Sidney and Steiner [15]. Let $P_0=(N_0,R_0)$ be a poset of m elements with $N_0=\{i_1,i_2,\dots,i_m\}$ and let $P_h=(N_h,R_h)$ for $h=1,2,\dots,m$ be disjoint posets. The *composition poset* $P=(N,R)$ is defined by $N=\bigcup_{h=1}^m N_h$ and $R=\bigcup_{h=1}^m R_h \cup \{(i,j): i \in N_h, j \in N_k \text{ and } \langle i_h, i_k \rangle \in R_0\}$. For this composition, a notation $P=P_0[P_1,K,P_m]$ is used, where P_0 is referred as *outer factor* and P_1,K,P_m as the *inner factors*. Then P is the *series composition* **S** of the inner factors when P_0 is a chain, and the *parallel composition* **P** when $i_h \sim i_k$ for all $i_h, i_k \in N_0$. In any other case, P is called a neighborhood composition **N**. Here, each inner factor is a module of P . Buer and Möhring [1] and Muller and Spinrad [13] propose algorithms for *canonical decomposition* of a poset into modules. For a poset $P=(N,R)$ these algorithms construct the *composition tree* $T(P)$ by identifying one of the following three possibilities:

(**P**) $P=P_0[P_1,K,P_m]$ ($m > 1$) is a parallel composition.

(S) $P = P_0[P_1, K, P_m]$ ($m > 1$) is a series composition.

(N) The maximal modules different from N partition N into $\prod_{h=1}^m N_h$ ($m \geq 4$) with $P = P_0[P_1, K, P_m]$ being a neighborhood composition, where P_1, K, P_m are the subsets of P induced by N_1, K, N_m respectively, and P_0 is the outer factor obtained from P by contracting each P_h into a new element i_h , $h = 1, K, m$.

By identifying for the original poset P which of these cases applies and by repeating this operation for each inner factor found, the modular decomposition procedure is implemented as iterative process until all factors have been decomposed into single elements. The composition tree $T(P)$ is a data structure to represent this process: the root of $T(P)$ is labeled by the set N and is assigned a type indicator S, P, or N, depending on which of the three cases applies to P . The sons of N correspond to the inner factors P_h of P and are labeled by the corresponding set N_h and type indicators S, P, or N, depending on which case applies to P_h . The leaves of $T(P)$ correspond to single elements of P . For details on decomposition process and constructing $T(P)$ see [1, 13, 15]. Note that algorithm [13] requires $O(n^2)$ time for the decomposition of a poset into modules.

When the objective functions of scheduling problems satisfy job module property which states that any optimal solution to a problem defined by a job module is consistent with at least one optimal solution for the entire problem, efficient algorithms can be used to solve the scheduling problems [12, 15]. These algorithms obtain optimal sequences by finding optimal subsequences for progressively larger modules and use efficient procedures [9, 13] for locating modules in a precedence network. Let us give formal definition for job module property and DP recursion property. An objective function f possesses the *job module property* [12, 15] if it satisfies the following condition: If N' is a job module of poset $P = (N, R)$ and s' is an optimal sequence for the problem on $P' = (N', R')$, where P' is a subset of P induced by N' , then there exists an optimal permutation s for P such that $s' = s|N'$, where $s|N'$ is the restriction of s to N' , i. e., the permutation induced by s on N' . DP approach uses the following *recursion property* of the objective function $f(N)$ over the initial sets $S \in N$: $f(S) = \min\{g(f(S - j, S, j)): j \in I(S)\}$, where $f(\emptyset) = 0$, g is a recursion formula, $S - j$ is shorthand notation for $S \setminus \{j\}$, and $I(S) = \{j: j \in S \text{ and } j \text{ has no successors in } S\}$.

If the recursion g can be computed in constant time for any given set of arguments and K denotes the number of initial subsets of poset $P = (N, R)$,

then the computation of $f(N)$ requires $O(Kw)$ time, where w denotes the maximum size of any set $I(S)$ and is called the *width* of P . The procedure to find an optimal permutation, after $f(N)$ has been calculated, requires $O(nw)$ time: the optimal sequence is generated in the reverse direction, starting from $S = N$ and identifying a job $j \in S$ for which the minimum was obtained in the recursion property; this j is placed last among the jobs in S , and S is replaced by $S - j$ until we obtain $S - j = \emptyset$. Since $K \leq O(n^w)$ [15], the DP algorithm for a fixed positive w requires at most $O(n^w)$ time to find an optimal sequence if $P \in C_w$, where C_w denotes the class of posets with width less than or equal to w .

Sidney and Steiner [15] propose a combination of decomposition algorithm [9, 10] and DP to find optimal solutions for the problems that have both the recursion and the job module properties. They define the class D_w of posets such that for any positive $w \geq 1$ the poset P belongs to D_w if and only if P can be built by a finite number of successive compositions of posets in which every outer factor is in C_w , i.e., every outer factor has a width $\leq w$. Then, D_w also contains C_w , and D_2 contains series-parallel posets. The combination of dynamic programming (DP) with modular decomposition proposed in [15] enables to enlarge the polynomially solvable classes of sequencing problems.

Coming back to our due date assignment and scheduling problem, we can denote it as $1|D_w, w \geq 2; C_j \leq d_j = p_j + q|\varphi(F, q)$ in case when precedence constraints belong to the class D_w of posets with $w \geq 2$. If F is either linear $F_{LIN} = \sum w_j C_j$ or exponential $F_{EXP} = \sum w_j \exp(-\gamma C_j)$ functions, then the problem under consideration possesses the DP recursion property and the job module property. DP recursion property for the functions F_{LIN} and F_{EXP} holds since the recursion formulas for F_{LIN} and F_{EXP} , respectively, can be written in the following way: $g(F_{LIN}(S - j), S, j) = F_{LIN}(S - j) + w_j \sum_{i \in S} p_i$ and $g(F_{EXP}(S - j), S, j) = F_{EXP}(S - j) + w_j \exp(-\gamma \sum_{i \in S} p_i)$. To show that

functions F_{LIN} and F_{EXP} possess the job module property, we can use the result by Monma and Sidney [12] who have shown that the following three conditions are sufficient for the job module property to hold:

1. *Strong adjacent sequence interchange property* which is equivalent for the function to be priority-generating and therefore holds for F_{LIN} and F_{EXP} .

2. *Strong series network decomposition (strong SND) property* which is valid for objective function F if the following condition holds for all permutations s and t of the same set: for all sequences u and v , $F(s) \leq F(t)$ if and only if $F(u, s, v) \leq F(u, t, v)$.
3. *Consistency property* which is valid for objective function F if there exists a relation π defined on all pairs of sequences satisfying the following property: for all permutations s and t of the same set, if $F(s) \leq F(t)$ then $s \pi t$.

The verification that strong SND and consistency properties hold for F_{LIN} and F_{EXP} follows directly from [11, 12]. As a result, problem $1|D_w, w \geq 2; C_j \leq d_j = p_j + q| \varphi(F, q)$ is solvable in polynomial time for the fixed w by algorithms proposed in [6] and, in particular, in $O(n^{m-1})$ time if each N-type outer factor of precedence constraints has cardinality $\leq m$ for a fixed constant m .

Acknowledgements

The research was partly supported by INTAS (Projects INTAS 00-217 and 03-51-5501) and ISTC.

References

1. Buer H, Möhring RH (1983) A fast algorithm for the decomposition of graphs and posets. Math Oper Res 8: 170-184
2. Gordon VS (1981) Some properties of series-parallel graphs (in Russian). Izv. Akademii Nauk BSSR (Proc Academy of Sciences of BSSR) 1: 18-23
3. Gordon VS, Proth J-M, Chu C (2002) A state-of-the-art survey of due date assignment and scheduling research: SLK, TWK, and other due date assignment models, Product Planning & Control 13: 117-132
4. Gordon VS, Shafransky YM (1977) On a class of scheduling problems with partially ordered set of jobs (in Russian). In: Proc 4th All-Union Conf Problems of Theoretical Cybernetics, Novosibirsk, pp 101-103
5. Gordon VS, Shafransky YM (1978) Optimal ordering with series-parallel precedence constraints (in Russian). Doklady Akademii Nauk BSSR (Reports Academy of Sciences of BSSR), 22: 244-247
6. Gordon VS, Strusevich VA (1999) Earliness penalties on a single machine subject to precedence constraints: SLK due date assignment. Comput & Oper Res 26: 157-177

7. Lawler EL (1978) Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals Discrete Math* 2: 75-90
8. Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB (1993) Sequencing and scheduling : algorithms and complexity. In: Graves S, Rinnooy Kan AHG, Zipkin P (eds) *Handbooks in Oper Res and Manag Sci. Vol. 4, Logistics of Production and Inventory*, North Holland, Amsterdam, pp 445-522
9. Möhring RH, Rademacher FJ (1984) Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals Discrete Math* 19: 257-356
10. Möhring RH, Rademacher FJ (1985) Generalized results on the polynomiality of certain weighted sum scheduling problems, *Methods Oper Res* 49: 405-417
11. Monma CL, Sidney JB (1979) Sequencing with series-parallel precedence constraints, *Math Oper Res* 4: 215-234
12. Monma CL, Sidney JB (1987) Optimal sequencing via modular decomposition: characterization of sequencing functions, *Math Oper Res* 12: 22-31
13. Muller JH, Spinrad J (1989) Incremental modular decomposition: Polynomial algorithms, *J ACM* 36: 1-19
14. Sidney JB (1975) Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Oper Res* 23: 238-298
15. Sidney JB, Steiner G (1986) Optimal sequencing by modular decomposition: Polynomial algorithms. *Oper Res* 34: 606-612
16. Tanaev VS, Gordon VS, Shafransky YM (1994) *Scheduling Theory. Single-Stage Systems*. Kluwer Academic, Dordrecht
17. Valdes JR, Tarjan E, Lawler EL (1982) The recognition of series-parallel digraphs., *SIAM J Comput* 11: 361-370

A Parallel Approach to the Pricing Step in Crew Scheduling Problems

T. V. Hoai¹, G. Reinelt², and H. G. Bock¹

¹ Interdisciplinary Center for Scientific Computing, University of Heidelberg
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

² Institute for Computer Science, University of Heidelberg
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

Abstract. When solving crew scheduling problems by column generation, the main task is to solve the pricing problem for introducing new columns. This problem is \mathcal{NP} -hard and usually requires more than 90% of the overall computation time in all of our experiments as well as in experiments reported in the literature. Therefore it is critical to achieve good performance in this step. This paper discusses an approach of using a cluster of computers to solve the pricing problem. Several aspects of parallelizing the pricing step are investigated and computational results are reported. The parallel algorithms are designed in such a way that they facilitate extensions and generalizations.

1 Introduction

The planning of airline operations usually consists of four phases. First, a timetable is constructed to satisfy the expected passenger transportation requests. Then aircrafts are allocated to flight legs. The third step is the crew scheduling problem, the problem we are dealing with in this paper. It consists of generating rotations (pairings) of crews in order to cover all flight legs. For the assignment of crews to flights a set of complicated airline rules and regulations according to laws have to be observed. Note that in this step, individuals are not yet considered. The final step called the rostering problem is the task of assigning pairings to individual crews where further aspects, like vacation plans or technical skills, are taken into account.

We model the crew pairing problem as an integer linear programming problem, or more precisely as a *set partitioning problem*

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = 1 \\ & x \in \{0, 1\}^n. \end{aligned}$$

In the context of the crew scheduling problem, a column of the matrix A corresponds to a feasible pairing and a row to a flight that has to be serviced. We have $A_{ij} = 1$ if pairing j covers flight i , and $A_{ij} = 0$, otherwise. The cost of pairing j is given by c_j . The variable x_j states whether the respective pairing is in the solution or not. The cost c_j of a pairing includes

the crew costs, the accommodation costs and the penalty costs. The particular property of this approach is that we have a problem with a huge number of variables, since virtually every possible pairing should be taken into account.

We solve the problem with a branch-and-cut algorithm based on the canonical linear programming relaxation where the condition $\mathbf{x} \in \{0, 1\}^n$ is replaced by $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$. The appropriate way to solve such linear programs is to employ column generation where one starts with a subset of possible pairings and adds further ones as needed. The central step of column generation is therefore similar to the pricing of nonbasic variables in the *revised simplex method*.

At each major iteration k of the column generation method, we have the current matrix \mathbf{A}^k , the objective vector \mathbf{c}^k , the variable vector \mathbf{x}^k , and the dual vector \mathbf{u}^k . After solving the so-called *restricted master problem*,

$$\begin{aligned} \min \quad & \mathbf{c}^{kT} \mathbf{x}^k \\ \text{s.t.} \quad & \mathbf{A}^k \mathbf{x}^k = \mathbf{1} \\ & \mathbf{0} \leq \mathbf{x}^k \leq \mathbf{1}, \end{aligned}$$

the solution $(\mathbf{x}^k, \mathbf{u}^k)$ will be used as input to the pricing step. The *pricing problem* amounts to solving the optimization problem

$$\min_{j \in \mathcal{J}} \mathbf{c}_j - \mathbf{u}^{kT} \mathbf{A}_{.j}$$

where \mathcal{J} is the set of all possible variables in the master problem. The purpose of the pricing step is to find negative reduced cost columns if they exist or to prove that there are no such columns.

The computation time for solving the pricing problem dominates the total time. In the context of the crew pairing problem, it is an \mathcal{NP} -hard problem and the time spent for solving the pricing problem usually amounts to about 90% of the total computation time. In this paper we cannot cover all aspects of the crew scheduling problem. Instead we only focus on how to solve the pricing subproblem effectively.

Many algorithms have been suggested to solve the pricing problem to optimality. Desrosiers et al. [4] discuss many aspects of using resource constrained shortest path algorithms. Another way is to use k -shortest path algorithms to find the most negative reduced cost variables. After ranking paths between two given nodes by k -shortest paths algorithms, a rule and regulation checking procedure will eliminate infeasible pairings. An application of a k -shortest path algorithm [6] is developed by the CARMEN system to solve the crew scheduling problem. Constraint Programming comes into play with the ability to model these rules quickly and easily. General approaches are discussed in [5] and there are many papers concerning the use of constraint programming in crew management operations.

There has already been effort to use parallel computers for solving crew scheduling problem. In PAROS [2], the parallelization is not implemented in

the process of solving the linear relaxation problem of the master problem. Instead, PAROS distributes the enumeration of pairings over the processors and their outputs will be fed into a set covering optimizer which is an iterative Lagrangean heuristic. The set covering optimizer was also parallelized, but not as successful as the enumeration phase. The parallel algorithm showed good performance when solving some real-world problems of Lufthansa. Another system employing parallel computing is RALPH of Marsten [1997], but, unfortunately, there is no published report available. A further idea for using parallel computers was suggested in [7]. Here the huge number of feasible pairings is generated in parallel and then used to construct constraint matrices of set partitioning problems. These problems are then solved by branch-and-cut approaches.

In this paper we want to apply parallelization for solving the pricing problem faster. In Section 2, we discuss several sequential pricing algorithms in order to determine the ones suitable to be parallelized. A master-slave model and implementation aspects for obtaining good performance are main points of Section 3. The advantages and disadvantages of constraint logic programming for our research are discussed in Section 4. Section 5 gives some conclusions.

2 Sequential Algorithms

Methods for solving the pricing problem can be classified into the categories

- resource constrained shortest path algorithms,
- k -shortest paths algorithms,
- exhaustive and heuristic enumeration,
- constraint logic programming.

The two first methods are based on graph algorithms. Although much work has been done in the area of parallel shortest paths, the theoretical worst case execution times of these parallel algorithms have the same bounds as those of the sequential ones [3]. These algorithms only work well on sparse or regular graphs which are efficiently partitioned into subgraphs having few boundary nodes. Unfortunately, this is not the case for the flight graphs of crew pairing problems. Moreover, there has been little effort for parallelizing resource constrained shortest paths and k -shortest paths algorithms on distributed memory machines. Most of them are concerned with shared memory systems. For example, in [8] an algorithm was suggested for the theoretical concurrent-read exclusive-write PRAM model. This is mainly due to the fact that these algorithms have a strong data dependency among computing nodes which is only inefficiently implemented on distributed memory parallel computers. Another disadvantage of these methods is that they are not well suited for extensions. Since airlines rules must be embedded into graphs, including

a new rule requires the reconsideration of the graph structure and algorithms as well. This difficulty also occurs in sequential algorithms.

Surprisingly, enumeration turned out to be a good choice for parallelization although it is the worst solution to many combinatorial optimization problems. The general implementation is usually easy because the rule checking can be treated separately as a black box. Moreover, we can apply application-specific heuristics to speed up the search process. A master-slave model can be used to implement an implicit exact search method. There is a job dispatcher to deliver dual values and lower bounds to enumeration blocks and receive pairings from them.

3 Parallel Pricing

Enumeration is parallelized in a straightforward way. The single jobs are independent, whether we use implicit (like constraint programming) or explicit (like exhaustive enumerating) approaches. They can be shared among processors of a cluster without any data dependency. The most effective parallel model for our problem is the master-slave model. In addition to the divide-and-conquer framework, we also use a bounding technique which is very helpful to reduce the search region.

For easy scalability, in our master-slave model, we dedicate processor 0 to be only responsible for distributing jobs, waiting for results and running the branch-and-cut kernel based on the framework ABACUS [9]. Note that the job of a slave in this approach consists of finding negative reduced cost pairings starting from a given flight. Due to the centralized control, we can easily apply a bounding technique. Remember that the initial lower bound is zero because we only find negative reduced cost pairings. We also use some application-specific heuristics to improve the performance of the algorithm.

Another point to be addressed is from which flight leg the enumeration will start in the next iteration of the column generation. If we always start from the same flight leg (e.g., the first flight of the schedule) in every iteration, we possibly only generate new pairings which are similar in structure to the pairings already contained in the current restricted master problem. Therefore they will only slightly improve our integer programming model. In order to make the search more effective, we change the starting flight leg in every iteration. In our implementation we start from the flight leg right after the one where the previous iteration stopped.

Since the computation time of each job is nondeterministic, we can experience an unfortunate behavior of the slaves. Namely, it can happen that, although the termination condition has been reached (e.g., enough new pairings have been generated or there is no more flight leg to send to slaves), the master still must wait for the completion of all slaves. If a slave has been assigned a difficult task, then the master also must suspend its further activities.

```

1: timer.start();
2: while( !( terminate condition reached ) ){
3:   forall p idle {
4:     if ( !( terminate condition reached ) ){
5:       send( p, SeqNo );
6:       send( p, job );
7:     }
8:   }

8:   if (( some slave idle ) && ( #pairings found > 0 )){
9:     FreeTime = timer.stop();
10:    if ( FreeTime / BusyTime >= IDLERATIO )
11:      break;
12:    else
13:      timer.start( without reset );
14:  }

15:  if ( incomming result from p ){
16:    recv( p, Slave_SeqNo );
17:    if ( SeqNo != Slave_SeqNo )
18:      continue;
19:    recv( p, result );
20:    if ( terminate condition reached ){
21:      BusyTime = timer.stop( with reset );
22:      timer.start( reset );
23:    }
24:  }
25: }
26: SeqNo ++;

```

Fig. 1. Number sequencing technique

We employ a numbering technique, outlined in Figure 1, to reduce the idle times of slaves between the solution of the current linear programming relaxation and the pricing step. Each computing job sent to slaves is accompanied by a sequence number. This number is compared with the number received from slaves (lines 17, 18). If they match, the received result is valid for the current iteration, otherwise, we discard the result. The code segment of lines 8–14 has the consequence, that if `FreeTime` (i.e., the duration between the time when the termination condition was reached and the current time) is large, then the algorithm will stop waiting and go immediately to the linear programming relaxation. The length of that period is controlled by the ratio `IDLERATIO`. This ratio should be kept small (e.g., 0.1%).

For our computational experiments, we generated several crew pairing problem instances at random, however taking the structure and regulations of real problems of Vietnam Airlines into account. The original instances turned

out to be too small and therefore too easy. All instances are guaranteed to have feasible solutions, each flight set has about 510 flight legs. The instances are very difficult for all sequential algorithms.

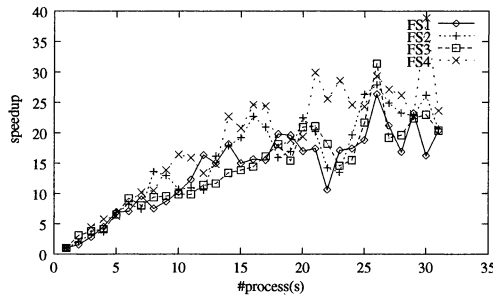


Fig. 2. The speedup of the master-slave approach

With the features we have discussed so far we obtain a good reduction of the computation time. In Figure 2, speedups are almost linear on our test instances (FS1–4), especially for a number of processors less than about 16. This is also due to the fact that our parallel approach distributes flight legs to slaves in the same order as in the sequential program. Therefore the search path remains more or less the same in the parallel run. The phenomenon that using many processors could give a bad performance in the figure is only due to the difficulty level of our test instances. We should only use more processors to solve more difficult instances. One of the interesting remaining questions is how to represent airlines rules and deal with them efficiently and quickly.

4 Parallel Pricing and Constraint Logic Programming

Clearly, it is more comfortable to use the declarative programming model of constraint programming to represent airlines rules. Our implementation uses ECLⁱPS^e[1]. Let \mathcal{F}_b be a flight set that is to be scheduled in an intended schedule period. Because the schedule is the same for next periods and the duration of a pairing is limited within the maximum number of days `MaxDays` we repeat \mathcal{F}_b several next periods in order that all possible pairings will be covered, obtaining a new flight set \mathcal{F} . This set will be used as a base domain for flight variables. If P is a pairing variable, it must be a list of flight variables whose domains are \mathcal{F} . Note that the first flight variable only has a domain of \mathcal{F}_b because the first flight must be within the schedule period. With these notations, we can easily present all airlines rules. In order to improve the performance of the algorithm, we also include a constraint to exclude pairings in advance which cannot have negative reduced costs. The idea behind this

is to reduce the domain of a flight variable of a pairing variable with the help of dual values and the minimum resting period between flights.

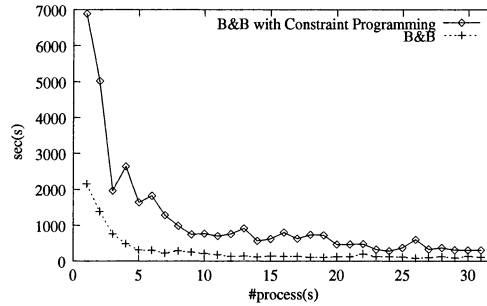


Fig. 3. The computation time of two pricings, one uses C++ rule checking and the other one constraint logic programming rule checking

Search on the finite domains of variables is a good target for parallelization. Domains could be decomposed independently into several sub-domains depending on computing resources. We again apply the master-slave model and the techniques mentioned in Section 3. For ECL^iPS^e , the first flight variable of a pairing variable now has a domain of one flight leg which is assigned by the master. All rules described by ECL^iPS^e remain unchanged. Therefore, it is easy to include more airline regulations. However, the computation times are quite bad mainly due to the slow search performed by ECL^iPS^e . Using up to 22 processors, we could only obtain a computation time of about 1 hour for the same problems which have been solved by the previous method. In view of this we decided to use constraint logic programming only for checking the validity of pairings. In this way, the implementation not only exploits the property of representing rules easily, but also inherits the efficiency of the branch-and-bound framework.

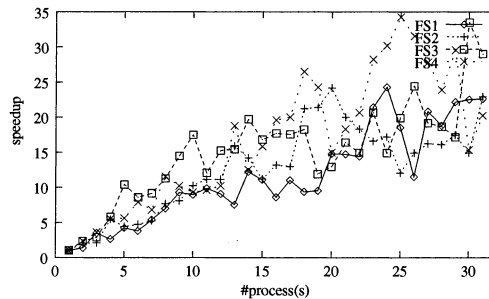


Fig. 4. Speedups to solve test instances with using constraint logic programming for rule checking

The computation time is now more reasonable. In Figure 3, the problem FS1 is solved within 2 hours. However, also in this figure, we see that the rule checking by constraint logic programming cannot have a performance as good as the C++ rule checking. As mentioned above, by reusing all features of the C++ implementation we also obtain good speedups in solving our test instances, depicted in Figure 4.

5 Concluding Remarks

Crew pairing problems are very hard and the computing time is mainly spent in the pricing part. Reducing the time for pricing significantly improves the branch-and-price approach for solving practical problems. In a fairly general framework, we have addressed some aspects of parallelizing sequential pricing algorithms and selected the best algorithmic approach to the crew pairing problem. Our parallel pricing algorithms can also handle further extensions, e.g., quickly changing rules and regulations of airlines by using constraint logic programming. The computation time for the pricing step decreases dramatically proportional to the number of processors in use. It will be helpful to apply our approach to solve the crew pairing problem with the branch-and-price framework.

References

1. A. Aggoun, D. Chan, P. Dufresne, E. Falvey, H. Grant, W. Harvey, A. Herold, G. Macartney, M. Meier, D. Miller, S. Mudambi, S. Novello, B. Perez, E. van Rossum, J. Schimpf, K. Shen, P. A. Tsahageas, and D. H. de Villeneuve. *ECLiPSe: User Manual, Release 5.2*. ECRC and IC-Parc, 2000.
2. P. Alefragis, C. Goumopoulos, E. Housos, P. Sanders, T. Takkula, and D. Wedelin. Parallel Crew Scheduling in PAROS. In *Proc. Euro-Par'98 Parallel Processing: 4th International Euro-Par Conference*, 1998.
3. B. Awerbach and R. S. Gallager. Communication Complexity of Distributed Shortest Path Algorithms. Technical report, MIT, 1985.
4. J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. *Handbooks in Operations Research and Management Science*, volume 8, chapter Time Constrained Routing and Scheduling, pages 35–139. North-Holland, 1993.
5. N. Guerinik and M. V. Caneghem. Solving Crew Scheduling Problems by Constraint Programming. In *Proceedings of the 1st Conference of Principles and Practice of Constraint Programming*, pages 481–498, 1995.
6. T. Gustafsson. *A Heuristic Approach to Column Generation for Airline Crew Scheduling*. PhD thesis, Chalmers University of Technology, 1999.
7. D. Klabjan and K. Schwan. Airline Crew Pairing Generation in Parallel. Technical report, The Logistics Institute, Georgia institute of Technology, 1999.
8. E. Ruppert. Finding the k Shortest Paths in Parallel. *Algorithmica*, 28:242–254, 2000.
9. S. Thienel. *ABACUS 2.0: User's Guide and Reference Manual*. University of Cologne, 1997.

Scheduling Regular and Temporary Employees with Qualifications in a Casino

Christoph Stark and Jürgen Zimmermann

{christoph.stark, juergen.zimmermann}@tu-clausthal.de

Clausthal University of Technology, Institute for Business Administration,
Julius-Albert-Str. 2, D-38678 Clausthal-Zellerfeld, Germany

Abstract. We consider a workforce scheduling problem that arises when scheduling the employees of a casino for a prescribed time period. Given a set of tasks for each day, we are looking for a sequence of tasks and days off for each employee. All sequences have to observe (hard) legal and contractual constraints as well as (soft) in-house restrictions. The overall objective is to minimize violations of the soft restrictions and evenly distribute attractive tasks among employees. Apart from regular employees, we have to take into account temporary employees, where each employee is characterized by a certain set of skills.

First, we consider an exact solution procedure based on a minimum-cost multi-commodity network flow formulation, which can be solved by means of column generation and branch-and-bound techniques. For large problem instances we propose a local search algorithm. An initial solution is obtained by successively solving several assignment problems. Two Hill Climbing procedures are used to improve the initial solution.

1 Introduction

Service industries are usually characterized by high personnel expenses as well as varying demands. Varying demands together with the fact that services cannot be stored lead to a changing labor utilization over time. Since employees cannot be hired or laid off easily in the short-term, peaks in the labor utilization have to be absorbed by temporary personnel or flexible work schedules.

In this paper, we consider a workforce scheduling problem arising in casinos. Increasing absence of customers throughout the past years forces casino carriers to pursue a policy of cost-cutting. Scheduling a workforce efficiently is an important possibility to cut down on personnel costs. Generating work schedules manually often requires the working time of one entire employee, while the application of automated scheduling systems dramatically decreases the required computation time and at the same time increases the quality of the generated schedules.

Based on a forecast of the number of expected customers, we assume that the kind and number of games to be offered is given. Every game belongs to an early (E), mid-day (M), late (L), or long late shift (LL) and can be subdivided into tasks (cf. Figure 1), each of which requires certain skills.

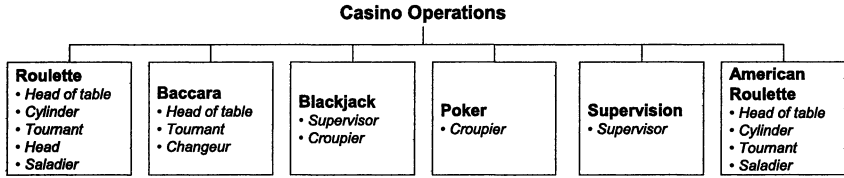


Fig. 1. Casino games and corresponding tasks

Each employee belongs to one of several wage scales. Associated with a certain wage scale is a set of skills that are required for a task. Figure 2 shows two screenshots from our implementation, namely the scheduling of games (cf. Figure 2(a)) and the definition of skills for an individual employee (cf. Figure 2(b)). The aim of workforce scheduling is to find a sequence of tasks and

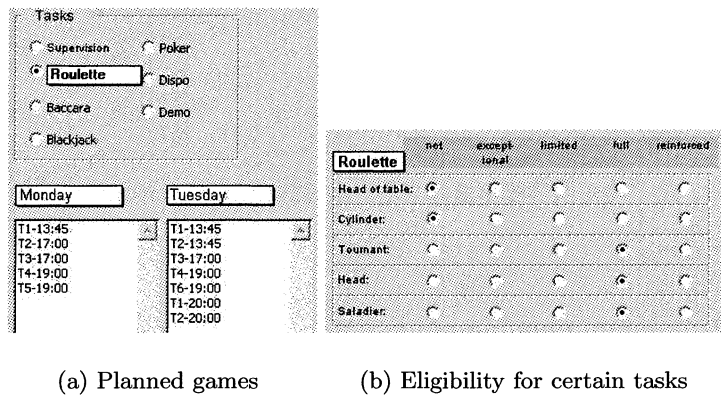


Fig. 2. Screenshots

days off (DO) for each employee. A sequence is feasible if the corresponding employee possesses the skills required for all tasks in the sequence and if it complies with additional restrictions given by (hard) legal and contractual constraints due to wage agreements. Moreover, we consider (soft) in-house restrictions, which serve to improve the working morale. Our objective is to minimize violations of soft restrictions and to evenly distribute attractive tasks among employees.

In Section 2 we specify the hard and soft restrictions as well as the objectives in detail. In Section 3.1 we present a minimum-cost multi-commodity network flow formulation for the problem under consideration. An exact solution can be obtained by means of a branch-and-price procedure. In Section 3.2 we propose a local search procedure for large problem instances. In a construction phase, an initial solution is obtained by iteratively solving several assignment problems. The initial solution is then improved by two Hill Climbing procedures. We close the paper with some conclusions.

2 Restrictions and Objectives

In a casino, workforce scheduling is subject to manifold restrictions. We have to distinguish between hard restrictions that must be obeyed and soft restrictions which should be obeyed. Since there may be no feasible solution which satisfies all soft restrictions, violations of soft restrictions are penalized.

1. Hard restrictions

- (a) No employee may be assigned to a task he is not qualified for.
- (b) Each employee must receive two days off for every 5 work days.
- (c) No employee may work longer than 10 days in a row.
- (d) In between two shifts, there must be at least 11 hours of idle time.

2. Soft restrictions

- (a) Every employee should be assigned at least one day off every week.
- (b) No employee should be assigned an isolated day off. Instead, at least two days off in a row should be scheduled.
- (c) A day off should be preceded by another day off or an early shift and should be succeeded by another day off or a (long) late shift.
- (d) Every employee should work in between 3 and 7 days in a row.
- (e) A vacation should be preceded by an early and succeeded by a (long) late shift.
- (f) No employee should work the same task more than 4 days in a row.

Nightwork as well as work on Sundays and holidays is awarded with extra pay of up to 190% of the ordinary wages. Thus, next to finding a feasible solution which minimizes violations of restrictions 2(a) – 2(f), we have to evenly distribute these boni, i.e., the corresponding attractive tasks, among all employees throughout the year.

3 Solution Methods

Often, workforce scheduling problems are formulated as set partitioning or set covering problems (cf. eg. [3]). However, the exponentially increasing number of decision variables makes even the linear relaxation of such problems hard to solve. Thus, we either need to (a) determine a way to efficiently handle large numbers of variables or (b) employ heuristics. In order to solve problems with a large number of variables, branch-and-price algorithms have been well proven in literature, for instance in the case of workforce scheduling problems arising in the airline industry (cf. e.g. [1] and [4]), retailing industry (cf. e.g. [7]), or health services (cf. e.g. [8]). Moreover, there are several heuristic approaches to be found, for instance Tabu Search (cf. e.g. [5]) or Simulated Annealing algorithms (cf. e.g. [2]). Approaches for problems subject to precedence constraints between tasks can be found in [9].

We outline an exact branch-and-price procedure for a simplified version of our workforce scheduling problem. Furthermore, we propose a local search procedure which is able to solve large problem instances with respect to all our restrictions and objectives.

3.1 Branch-and-Price

We consider a planning horizon of $h = 1, \dots, 14$ days. S_h denotes the set of all tasks on day h . With each task $i \in S := \bigcup_{h=1}^{14} S_h$ we associate a pair (s_i, e_i) representing the events “start” and “end” of task i . The set of pairs (s_i, e_i) for all $i \in S$ is denoted by S' . So called transitions are used to model changeovers from one task to another. A transition between two tasks $i \in S_h$ and $j \in S_{h'}$, $h < h'$, is modelled as a pair (e_i, s_j) . T denotes the set of all transitions.

In what follows, we consider a network $G := (N, A)$. The set of nodes N consists of all s_i and e_i ($i \in S$) as well as a source r and a sink s . T' consists of the pairs (r, s_i) for all $s_i \in N$, (e_i, s) for all $e_i \in N$, $i \in S$, and (r, s) . For each pair $(i, j) \in S' \cup T \cup T'$ graph G contains a directed arc (i, j) between nodes i, j . Thus, the set of arcs A can be represented by $S' \cup T \cup T'$.

For each wage scale $q \in \{1, \dots, Q\}$ we have a set of employees K^q . Each task $(i, j) \in S'$ has to be assigned to an employee $k \in K^q$ in between a minimum and maximum required wage scale $\underline{q}_{ij}, \bar{q}_{ij} \in \{1, \dots, Q\}$, respectively. With each arc $(i, j) \in T \cup T'$ we associate a weight c_{ij} and w.l.o.g. we assume that this weight is equal for all employees $k \in K^q$, $q \in \{1, \dots, Q\}$. Weights c_{ij} are used to penalize violations of restrictions 1(d), 2(c), and 2(e) (cf. Section 2). With P^q we denote the set of all feasible paths from r to s in G for an employee $k \in K^q$. Such a path can be interpreted as a sequence of tasks and days off. Path $p \in P^q$ is feasible if $\underline{q}_{ij} \leq q \leq \bar{q}_{ij}$ for all arcs $(i, j) \in S'$ on path p . Moreover, we introduce a binary variable δ_{ij}^p which equals 1 if $(i, j) \in A$ is part of path $p \in P^q$, $\delta_{ij}^p = 0$ otherwise. $c(p)$ denotes the length of path $p \in P^q$, i.e., $c(p) = \sum_{(i,j) \in A} \delta_{ij}^p c_{ij}$. Finally, we require a set of decision variables φ_p^k which equal 1 if employee $k \in K^q$ is assigned path $p \in P^q$, $\varphi_p^k = 0$ otherwise. A minimum-cost multi-commodity network flow formulation for our workforce scheduling problem with the objective of minimizing violations of restrictions 1(d), 2(c), and 2(e) and subject to restrictions 1(a), 1(b), and 1(c) (cf. Section 2) now reads

$$\text{Min.} \quad \sum_{q=1}^Q \sum_{p \in P^q} \sum_{k \in K^q} c(p) \varphi_p^k \quad (1)$$

$$\text{s.t.} \quad \sum_{p \in P^q} \varphi_p^k = 1 \quad \forall k \in K^q, q \in \{1, \dots, Q\} \quad (2)$$

$$\sum_{a=\underline{q}_{ij}}^{\bar{q}_{ij}} \sum_{k \in K^a} \sum_{p \in P^a} \delta_{ij}^p \varphi_p^k = 1 \quad \forall (i, j) \in S' \quad (3)$$

$$\sum_{p \in P^q} \left(\sum_{(i,j) \in S'} \delta_{ij}^p \right) \varphi_p^k \leq 10 \quad \forall k \in K^q, q \in \{1, \dots, Q\} \quad (4)$$

$$\varphi_p^k \in \{0, 1\} \quad \forall k \in K^q, q \in \{1, \dots, Q\}, p \in P^q \quad (5)$$

One commonly used possibility to solve the latter problem is the application of a branch-and-price procedure. In this context, the efficient solution of the linear relaxation of the underlying problem is of particular importance and can be achieved by means of a column generation approach. Since the linear relaxation Φ of problem (1) – (5) generally contains a very large number of decision variables, we initially consider a restricted version Φ' of problem Φ , called the *restricted master problem*. Φ' contains only a small subset of the variables φ_p^k . A proper subset of such variables can be derived from any basic feasible solution of Φ . Such a solution can in turn be obtained by generating a disjunctive path for each employee from r to s subject to the hard restrictions described in Section 2.

Starting with an optimal solution to Φ' , in each step we determine some additional variable φ_p^k that is appended to Φ' . The problem of finding a variable φ_p^k with minimum reduced objective function coefficient $\tilde{c}(p) < 0$ is called the *sub-problem*. For the problem at hand, this sub-problem is equal to the solution of several shortest path problems in G , one for each $k \in K^q$, $q = 1, \dots, Q$.

Given an optimal solution to Φ' and the values α_k , β_{ij} , and γ_k of the dual variables for each restriction (2), (3), and (4), respectively, the reduced objective function coefficient of any variable can be computed as

$$\tilde{c}(p) = c(p) - (\alpha^T, \beta^T, \gamma^T) a_p \quad ,$$

where a_p is the column in the coefficient matrix corresponding to variable φ_p^k . a_p is of the form $\left(1^{kq}, \delta^p, 1^{kq} \left(\sum_{(i,j) \in S'} \delta_{ij}^p \right) \right)^T$, where 1^{kq} is the $\sum_{a=1}^{q-1} |K^a| + k$ -th column of the $\sum_{q=1}^Q |K^q| \times \sum_{q=1}^Q |K^q|$ identity matrix. Pivoting an additional variable φ_p^k into the basis improves the current objective function value if

$$\sum_{(i,j) \in A \setminus S'} c_{ij} \delta_{ij}^p + \sum_{(i,j) \in S'} \delta_{ij}^p \underbrace{(c_{ij} - \beta_{ij} - \gamma_k)}_{\text{modified weight of arc } (i,j) \in S'} < \alpha_k \quad (6)$$

holds. Thus, for each employee $k \in K^q$ we determine a feasible shortest path from r to s with respect to the modified weight $c_{ij} - \beta_{ij} - \gamma_k$ of the arcs $(i,j) \in S'$. If the resulting total length of the shortest path is less than α_k , then the path is appended to problem Φ' , i.e., we add another variable φ_p^k as well as the corresponding column a_p to Φ' .

The process of solving Φ' and finding and appending shortest paths is repeated until there is no shortest path for any employee $k \in K^q$ for which condition (6) holds. The proposed column generation approach can be embedded into an appropriate branch-and-bound framework in order to obtain integral solutions.

3.2 Local Search

Generating an Initial Solution An initial solution is obtained by determining for each day an assignment between employees and tasks. Thus, we formulate an assignment problem for every day, which can be visualized as a bipartite graph (cf. Figure 3). Source nodes correspond to available employees and sink nodes correspond to daily tasks as well as days off. The number of days off an employee is entitled to depends on the number of days he is available for work. The total number of days off is allocated to individual days according to a predefined strategy. Moreover, sources and sinks are fully interconnected by directed arcs, which represent an assignment of employees to tasks or days off.

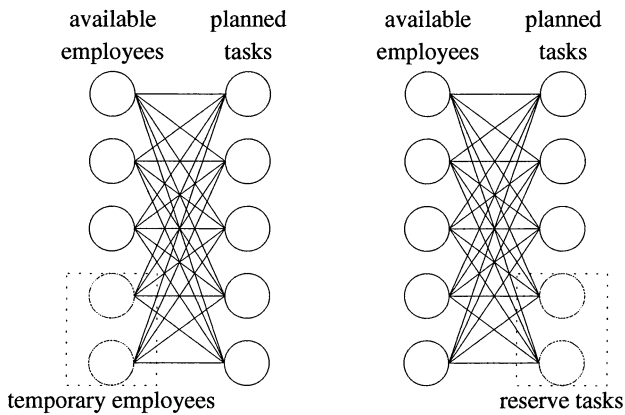


Fig. 3. An assignment problem for a single day

To ensure that the number of sources is equal to the number of sinks, we insert either further (temporary) employees or (reserve) tasks (cf. Figure 3).

The most important step during the initialization of a daily assignment problem is the determination of appropriate arc weights. Our goal is to determine arc weights which lead to work sequences in accordance with the underlying restrictions 1(a) – 2(f). A good work sequence should resemble a sequence of shifts of the form (LL, ..., L, ..., M, ..., E, ..., DO, ..., LL, ...). We consider three factors when determining arc weights:

- *Qualification:* Arc weights are used to penalize assignments of employees to tasks for which they are not sufficiently skilled.
- *Previous assignments:* We consider the number of uninterrupted work days, the shift types of the last three assignments, and the shift type of the assignment under consideration. Long periods of work as well as illegal or unfortunate work sequences are penalized.
- *Day off assignments:* We consider the number of days off an employee has yet to receive, the remaining number of days until the next predefined

day off or the end of the planning horizon, and the shift types of previous assignments. Supernumerous and insufficient assignments of days off as well as predictably unfortunate assignments are penalized.

The assignment problem for each day can be polynomially solved by means of the well-known Glover–Klingman algorithm (cf. [6]). Note that the optimal solution of an assignment problem depends on solutions that were obtained for assignment problems on previous days.

Hill Climbing Procedure The initial solution is improved by two Hill Climbing procedures that pursue two hierarchically ordered objectives: (a) improving the quality of sequences and (b) evenly distributing boni (i.e., attractive tasks) among employees. Neighboring solutions are obtained by swapping the assignments between two employees for one and the same day of the planning horizon. Swapping is permitted only if the resulting neighboring solution complies with the hard restrictions in Section 2.

The first Hill Climbing procedure serves to improve the quality of individual work sequences regarding their violation of the soft restrictions 2a – 2f. Starting with $t = 14$, we swap two assignments on day $t - 1$, if this improves the objective function value of the assignment problem for day t . This step is repeated for $t = 13, \dots, 2$.

The second Hill Climbing procedure pursues the objective of evenly distributing boni for work on weekends, public holidays, etc. among all employees throughout the entire year. For this reason we minimize the absolute deviation of an employee's ratio of regular rate hours to bonus hours from the mean ratio over all employees. For every day $t = 1, \dots, 14$, we thus swap two assignments in t if this decreases the aforementioned deviation.

The chosen neighborhood operator ensures that, depending on the initial allocation of days off, a globally optimal solution can be reached from any starting solution. Moreover, it should be noted that our implementation of the local search algorithm contains further interesting features: (a) Employees can express wishes concerning single assignments, (b) if no feasible solution can be found, the algorithm completes its calculation and assists a human scheduler in developing an appropriate workaround, and (c) temporary employees are assigned to days and shifts for which they are actually available by means of a network flow problem.

4 Conclusion

Preliminary results show that the column generation approach (cf. Section 3.1) is able to solve small and medium sized problem instances. A proper branch-and-bound framework is currently being implemented. Moreover, it has to be investigated whether the proposed minimum-cost multi-commodity network flow formulation can be expanded in order to consider further practical restrictions.

The local search algorithm described in Section 3.2 has been implemented in the context of several diploma theses and successfully deployed with the “Spielbank Baden–Baden”. Concerning the fulfillment of the soft restrictions described in Section 2, solutions obtained by the algorithm have been at least as good as those obtained by an experienced human scheduler. However, while the working time of one entire employee was necessary to generate a work schedule for each employee, the local search algorithm generates a solution in less than two minutes. This enables users to generate several alternative work schedules. Reviewing the results takes about one hour and is facilitated by a reporting component, which points the user to unfortunate sequences. Moreover, a human scheduler was not able to pursue the objective of minimizing the deviation of an employee’s ratio of regular rate hours to bonus hours from the mean ratio over all employees. The proposed local search procedure is able to keep this deviation within 10%. Nevertheless, we feel that there is still some promising research to be done regarding the proposed local search approach. One major topic is the development of neighborhood operators which additionally allow to optimize the allocation of days off to individual days of the planning horizon.

References

1. Barnhart, C., Johnson, E.L., Anbil, R., Hatay, L. (1994): A Column–Generation Technique for the Long–Haul Crew Assignment Problem. In: Ciriani, T.A., Leachman, R.C. (eds.) *Optimization in industry 2*. Wiley, New York
2. Brusco, M.J., Jacobs, L.W. (1993): A Simulated Annealing Approach to the Cyclic Staff–Scheduling Problem. *Naval Research Logistics* 40, 69–84
3. Dantzig, G.B. (1954): A Comment on Edie’s “Traffic Delays at Toll Booths”. *Operations Research* 2, 339–341
4. Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M.M., Soumis, F. (1997): Crew pairing at Air France. *European Journal of Operational Research* 97, 245–259
5. Dowsland, K.A., Thompson, J.M. (2000): Solving a Nurse Scheduling Problem with Knapsacks, Networks and Tabu Search. *Journal of the Operational Research Society* 51, 825–833
6. Glover, F., Glover, R., Klingman, D. (1986): Threshold Assignment Problem. *Mathematical Programming Study* 26, 12–37
7. Haase, K. (1999): Modellgestützte Personaleinsatzplanung im Einzelhandel. *Zeitschrift für Betriebswirtschaft* 69, 233–244
8. Jaumard, B., Semet, F., Vovor, T. (1998): A Generalized Linear Programming Model for Nurse Scheduling. *European Journal of Operational Research* 107, 1–18
9. Salewski, F. (1999): Modellierungskonzepte zur Dienstplanung bei flexibler Personalkapazität. *Zeitschrift für Betriebswirtschaft* 69, 319–345

Web Robot Detection - the Influence of Robots on Web Mining

Christian Bomhardt and Wolfgang Gaul

Institut für Entscheidungstheorie und Unternehmensforschung
University of Karlsruhe (TH)
76128 Karlsruhe, Germany

Abstract. Web usage mining relies on web server logfile data. Parts of this data originate from web robots. This can - with respect to the original aims of web mining - lead to contradicting decisions based on distorted results. We describe possibilities of web robot detection and give examples how, e.g., e-metrics and results of association rule algorithms can differ based on raw logfiles versus those that consist of requests of human users or web robots.

1 Introduction

1993 is the year where some of the first known web robots appeared in the internet [24], e.g., "Wanderer" by Matthew Gray (measuring web growth) and "JumpStation" by J. Fletcher (indexing). At that time, commercial web usage played a minor role, instead overloaded web servers or waste of bandwidth were areas for robot deployment problems. Nevertheless, those problems together with a growing number of newly used robots led to a standard for robot exclusion ([18]). Today, most robots adhere more or less strictly to existing guidelines for robots. With about 16% of the web traffic originating from robots ([19]), nowadays, robot detection must be considered for serious web mining efforts. While cooperative robots can be detected with the help of a simple heuristics (as we will see later) malignant robots ignore the guidelines mentioned above and may even apply stealth technologies. Generally, there is not much known about malignant robots as their usage on the net is somehow "unethical" (e.g., extraction of mail addresses for spamming ([17]), unauthorized usage of US Government's Weather Service ([3])).

There exist four major categories of widely used robot detection technologies: *Simple methods*, *traps*, *web navigation behavior analysis*, and *navigational pattern modeling*. Simple methods check the [request], [agent], and [IP address] fields in webserver logfile entries against lists of known robot identifications ([5]). Traps consist of links within the HTML pages that are invisible for a human user. If such a link is visited, it must have been visited by a robot ([20]). Web navigation behavior analysis searches for typical navigation characteristics based on the objectives of the different classes of robots ([2]). Navigational pattern modeling calculates session attributes and applies

data mining algorithms that try to detect robots on the basis of selected session attributes ([22], [23], [10]). Malignant robots can be detected via traps, web navigation behavior analysis, and navigational pattern modeling. Simple methods are unlikely to detect malignant robots which are relatively sparse in the data. Thus, their detection requires additional efforts. This is why we concentrate on cooperative robots in this paper. With the increasing number of robots and changes of the identification information of known ones, the problem of keeping robot lists current and complete becomes more and more laborious. We faced this problem with the development of the robot detection tool RDT ([8]) - a specialized web logfile preprocessing software enabling researchers to effectively work with and understand large logfile data - one of the main requirements to effectively remove robot requests from web usage data and to accomplish further web mining steps.

In the following the *Web Data Preprocessing* process will be divided into the substeps sessionizing, and robot detection. Both steps are supported by the RDT software. It speeds up preprocessing and therefore enables researchers to focus on their specific mining tasks.

2 Web Data Preprocessing

Every webserver records served HTTP-requests in its logfile. In the following, the wide spread combined logfile format ([4]) is used, which most HTTP servers can create. This format contains the following nine fields: [IP address] as client IP address, [name] as name of the user (usually unused), [login] as login-name of the basic HTTP-authentication, [date] as date and time of the request, [request] as HTTP-request containing the request method, the URL of the requested resource (page), and the desired HTTP-protocol, [status] as 3-digit status code returned by the server, [size] as number of bytes actually returned by the server, [referrer] as URL of the referencing page and [agent] as name of the client agent. HTTP-requests are the basis for web usage mining ([13], [14]).

2.1 Sessionizing

Each HTTP-request is written in the order of occurrence into the server logfile and the construction of contiguous user requests requires further efforts. An overview of different types of sessionizers and their performance is given in [7]. In [13] a navigational path construction algorithm as basis for establishing sessions is described. For this paper, we selected a widely used heuristics from [12] where requests with the same agent and IP address are grouped together as long as the maximum idle time between two requests is smaller than 30 minutes. A common user session consists of two kinds of requests: main requests as result of an user action and auxiliary requests. Auxiliary requests are automatically issued by browsers to retrieve objects referenced

by the main request (e.g., stylesheets, images, background sound). If possible, we try to assign auxiliary requests - on basis of the referrer field - to their main requests.

2.2 Robot Detection

Sessions are seen as sessions of human users until hints for the opposite are found. Sessions with requests for files that are known to be never requested by human users (e.g., `robots.txt`, some hidden linked files from traps ([20] or typical files from worm attacks (e.g., `cmd.exe` for Nimbda ([16]))) provide good hints for robot detection. All these files are stored in the *trapfile list*. Robots that obey to the robot exclusion standard identify themselves with an agent tag that can be recognized as the agent of a robot. Those tags are collected in the *robot agent list*. Some IP addresses are known to be solely used by robots (e.g., the google bots) and can be saved in the *robot IP list*. The composition of these lists can be simplified by assimilating the list of known robots available from <http://www.robotstxt.org/wc/robots.html> ([24]). Figure 1 summarises what could be called robot detection heuristics.

```
function IsRobotSession( Session )
{
if (session contains a request of a file from the trapfile list)
    then return TRUE;
if (session agent is contained in the robot agent list)
    then return TRUE;
if (session IP is contained in the robot IP list)
    then return TRUE;
return FALSE;
}
```

Fig. 1. Robot detection heuristics

The magnitude of logfile data requires the application of web mining tools. Specialized software like our RDT can support researchers with respect to data understanding and functions of data preprocessing. Because of space restrictions properties of RDT have to be described in a different paper.

3 Empirical Results

We tried to find out whether robots influence web mining results. The logfile used in this paper originates from a mid-sized online shop. The whole dataset (all sessions), the part without robots (user sessions, 73%), and the part without users (robot sessions, 27%) were examined and compared. First, we selected some simple but wide-spread features (e.g., top N, top entry, top exit

pages) which are used by many common logfile analysing tools like webalizer ([6]). Second, we calculated selected e-metrics from [21] and micro conversion rates from [15] for the three datasets. Third, as a more specific application, we looked for association rules ([1],[13],[14]) within the different datasets and compared the results.

The results of the simple features are summarized in table 1 (Top N list of most frequently requested pages), table 2 (Top entry list of most frequently requested entry pages), and table 3 (Top exit list of most frequently used exit pages). The shadowed outcomes show that 7 positions out of the first 16 highest ranked pages of the top N list are affected by robots. A similiar, even stronger effect can be observed in the top entry list (11 differences out of the first 16 positions) and also in the top exit list (6 differences out of 16). Results of this kind can be used for improvements of the website.

Table 1. Top N list of most frequently requested pages

URL of page	All sessions Position	User sessions Position	Robot sessions Position
/	1	1	1
/shop/show/de	2	2	5
/shop/show/show_basket.php3	3	3	3
/shop/templates/basket.php3	4	4	11
/shop/show/show_search.php3	5	5	65
/zaubertricks.htm	6	6	17
/de/termine.htm	7	13	2
/de/gewinnspiel.htm	8	7	6
/de/index.htm	9	10	4
/shop/templates/order_adrform.php3	10	8	-
/shop/templates/order_finish.php3	11	9	-
/de/information.htm	12	11	7
/de/kontakt.htm	13	12	8
/shop/images/tr.gif/	14	14	-
/admin/onedit/fr_leer	15	15	-
/de/newsletter.htm	16	16	10

The influence of robots on the generic e-metrics hit-to-visit(1) (percentage of sessions with more than 1 request), hit-to-visit(2) (percentage of sessions with more than 2 requests), and avg. visit depth (number of pageviews) is rather small in contrast to the strong distortion with respect to the avg. visit duration (cp. table 4). The influence of robots on micro conversion rates as visit-to-basket (percentage of visits with basket usage), basket-to-buy (percentage of sessions with basket usage that lead to purchases), and visit-to-buy (percentage of visits with purchases) is small but relevant (cp. table 5). Here, one can see that robots never buy.

Table 2. Top entry list of most frequently requested entry pages

URL of entry page	All sessions Position	User sessions Position	Robot sessions Position
/	1	1	1
/shop/show/de	2	2	4
/zaubertricks.htm	3	3	12
/de/gewinnspiel.htm	4	5	5
/de/index.htm	5	8	3
/shop/show/show_basket.php3	6	6	16
/shop/show/show_search.php3	7	4	148
/shop/templates/order_adrform.php3	8	7	-
/de/information.htm	9	10	7
/kartentricks.htm	10	9	32
/de/termine.htm	11	12	2
/de/kontakt.htm	12	11	6
/de/newsletter.htm	13	13	9
/de/links.htm	14	17	13
/de/agb.htm	15	19	11
/taschenspielertricks.htm	16	14	46

Table 3. Top exit list of most frequently used exit pages

URL of exit page	All sessions Position	User sessions Position	Robot sessions Position
/	1	1	1
/shop/show/de	2	2	3
/zaubertricks.htm	3	3	14
/shop/show/show_basket.php3	4	4	18
/shop/show/show_search.php3	5	5	60
/de/gewinnspiel.htm	6	6	7
/de/kontakt.htm	7	7	6
/de/index.htm	8	8	4
/de/termine.htm	9	12	2
/kartentricks.htm	10	9	34
/shop/images/tr.gif/	11	10	-
/de/information.htm	12	11	5
/de/agb.htm	13	13	10
/de/links.htm	14	14	12
/de/newsletter.htm	15	16	9
/de/wir_ueber_uns.htm	16	18	13

Table 4. Results of e-metrics analysis

Selected e-metrics	All sessions	User sessions (73%)	Robot sessions (27%)
Number of sessions	52295	38227	14068
Hit-to-visit(1)	37363 (71%)	29728 (77%)	7635 (54%)
Hit-to-visit(2)	29700 (56%)	24971 (65%)	4729 (33%)
Avg. visit depth	16	16.6	14.5
Avg. visit duration	580s	236s	1512s

Table 5. Results of micro conversion rate calculations

Micro conversion rate	All sessions	User sessions	Robot sessions
Visit-to-basket	3.5%	4.2%	0.7%
Basket-to-buy	31.4%	32.7%	0%
Visit-to-buy	1.0%	1.4%	0%

Table 6. Selected results of association rule mining

	All sessions	User sessions	Robot sessions
Number of rules found	9	19	3
Number of interesting rules found	4	6	3
Association rule	(Support,Confidence)		
{/shop/show/de} ⇒ {/}	(30.1%,58.7%)	(37.6%,59.4%)	-
{/zaubertricks.htm} ⇒ {/}	(10.2%,68.1%)	(12.8%,69.8%)	-
{/de/information.htm} ⇒ {/shop/show/de}	(3.1%,65.5%)	(3.6%,62.9%)	-
{/de/information.htm, /shop/show/de} ⇒ {/}	(3.1%,65.3%)	(3.5%,64.3%)	-
{/shop/show/show_search.php3} ⇒ {/shop/show/de}	-	(3.7%,50.5%)	-
{/de/gewinnspiel, /shop/show/de} ⇒ {/}	-	(3.0%,64.5%)	-
{/de/newsletter.htm} ⇒ {/de/gewinnspiel.htm}	-	-	(3.0%,53.1%)
{/de/newsletter.htm} ⇒ {/de/index.htm}	-	-	(3.0%,52.2%)
{/de/newsletter.htm} ⇒ {/de/kontakt.htm}	-	-	(3.0%,51.3%)

AprioriPre ([9]) and apriori ([11]) were used for association rule mining. Support was set to 3% and confidence to 50% to limit the number of rules. Table 6 displays the rules which were found. Uninteresting rules reproduce facts already known ($\text{order_finish} \Rightarrow \text{show_basket}$ is such a rule if one cannot purchase without usage of the basket). Based on the selected support and confidence parameters the underlying dataset contained 4 interesting rules for all sessions. Two additional rules were found when only user sessions were evaluated. For the robot dataset completely different rules would have to be taken into consideration.

4 Conclusions and Outlook

Empirical results show that robots can seriously influence web mining. Some new aspects and distortions were discovered thanks to our robot detection efforts. The developed robot detection tool RDT can simplify and accelerate the logfile preprocessing step. It enabled us to effectively remove robot activities in large logfiles and it facilitated data understanding. The problem of detecting malignant robots will be addressed in a forthcoming paper.

References

1. Agrawal, R., Srikant, R. (1994) Fast Algorithms for Mining Association Rules, Proc. 20th Int. Conf. Very Large Data Bases, VLDB
2. Almeida, V., Riedi, R., Menascé, D., Meira, W., Ribeiro, F., Fonseca, R. (2001) Characterizing and Modeling Robot Workload on E-Business Sites, Proc. 2001 ACM Sigmetrics Conference, <http://www-ece.rice.edu/~riedi/Publ/RoboSim01.ps.gz>
3. Anaconda Partners LLC: Anaconda! Foundation Weather, http://anaconda.net/ap_wxdemo.shtml
4. Apache HTTP Server Documentation Project: Apache HTTP Server Log Files Combined Log Format, <http://httpd.apache.org/docs/logs.html\#combined>
5. Arlitt, M., Krishnamurthy, D., Rolia, J. (2001) Characterizing the Scalability of a Large Web-Based Shopping System, ACM Transactions on Internet Technology, <http://www.hpl.hp.com/techreports/2001/HPL-2001-110R1.pdf>
6. Barrett, B. (2001) Webalizer, <http://www.mrunix.net/webalizer/>
7. Berendt, B., Mobasher, B., Spiliopoulou, M., Wiltshire, J. (2001) Measuring the Accuracy of Sessionizers for Web Usage Analysis, Proceedings of the Web Mining Workshop at the First SIAM International Conference on Data Mining, Chicago
8. Bomhardt, C. (2002) The Robot Detection Tool, <http://www.bomhardt.de/bomhardt/rdt/produkt.html>
9. Bomhardt, C. (2003) AprioriPre, <http://www.bomhardt.de>
10. Bomhardt, C., Gaul, W., Schmidt-Thieme, L. (2003) Web Robot Detection - Preprocessing Web Logfiles for Robot Detection, Working paper, Institut für Entscheidungstheorie und Unternehmensforschung

11. Borgelt, C. (2003) Apriori, a Program to Find Association Rules With the Apriori Algorithm, <http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html>
12. Catledge, L., Pitkow, J. (1995) Characterizing Browsing Strategies in the World-Wide Web, Computer Networks and ISDN Systems
13. Gaul, W., Schmidt-Thieme, L. (2000) Frequent Generalized Subsequences - A Problem from Webmining, in: Gaul, W., Opitz, O., Schader, M. (eds.): Data Analysis, Scientific Modelling and Practical Application, Springer, Heidelberg, 429-445
14. Gaul, W., Schmidt-Thieme, L. (2002) Recommender Systems Based on User Navigational Behavior in the Internet, Behaviormetrika, Vol. 29, No.1, 1-22
15. Gomory, S., Hoch, R., Lee, J., Podlaseck, M., Schonberg, E. (2000) E-Commerce Intelligence: Measuring, Analyzing, and Reporting on Merchandising Effectiveness of Online Stores, Working paper, IBM T.J. Watson Research Center
16. Heng, C. Defending Your Web Site / Server From the Nimbda Worm / Virus <http://www.thesitewizard.com/news/nimbdaworm.shtml>
17. Ipaopao.com software Inc. Fast Email Spider For Web, <http://software.ipaopao.com/fesweb/>
18. Koster, M. (1994) A Standard for Robot Exclusion, <http://www.robotstxt.org/wc/norobots-rfc.html>
19. Menascé, D., Almeida, V., Riedi, R., Ribeiro, F., Fonseca, R., Meria, W. (2000) In Search of Invariants for E-Business Workloads, Proceedings of ACM Conference on Electronic Commerce, Minneapolis, MN, <http://www-ece.rice.edu/~riedi/Publ/ec00.ps.gz>
20. Mullane, G.S. (1998) Spambot Beware Detection, <http://www.turnstep.com/Spambot/detection.html>
21. NetGenesis (2000) E-Metrics: Business Metrics for the New Economy, NetGenesis Corp.
22. Tan, P.-N., Kumar, V. (2000) Modeling of Web Robot Navigational Patterns, Proc. ACM WebKDD Workshop, 2000
23. Tan, P.-N., Kumar, V. (2001) Discovery of Web Robot Sessions Based on their Navigational Patterns, <http://citeseer.nj.nec.com/cache/papers/cs/22262/http://zSzzSzwww.cs.umn.eduSz~ptanzSzdmkd.pdf/tan02discovery.pdf>
24. The Web Robots Pages www.robotstxt.org/wc/robots.html

Visualizing Recommender System Results via Multidimensional Scaling

Wolfgang Gaul¹, Patrick Thoma¹, Lars Schmidt-Thieme², and Sven van den Bergh³

¹ Institut für Entscheidungstheorie und Unternehmensforschung,
Universität Karlsruhe (TH)

² Institut für Informatik, Universität Freiburg

³ Mentasys GmbH, Karlsruhe

Abstract. Web site visitors who look for desired items can formulate search queries which are taken by recommender systems to provide support within the underlying buying situation (e.g., enabling users to view recommended items and buy the ones they find most appropriate). Data from a large German retail online store is used to visualize products viewed most frequently together with search profiles that represent identical search queries of larger subgroups of site users. Comparisons between products viewed most frequently and those purchased most frequently can be used to improve the generation of recommendations. The results give interesting insights concerning the searching, viewing, and buying behavior of online shoppers.

1 Introduction

“Recommender Systems” is the label for a methodology that can, e. g., be designed (among others) to guide online users through complex web sites, huge online-stores or any other kind of information that can not be overviewed or searched through completely. As this is only one of the many situations in which recommender systems can be applied a framework that can be used to classify such systems would be of help [2]. From an empirical point of view, we analyzed data gathered from a large German retail online store where a rule-based recommender system for a special product class was offered to support potential buyers. Customers received a sorted list of items of the underlying product class that “best” matched their search queries. We conducted multidimensional scaling (MDS)[1] to visualize the different search queries together with recommendable items of the product domain. This allowed us to analyze the searching behavior of site visitors on the basis of the most frequently chosen search queries, to compare user preferences with respect to the most appropriate products displayed within the MDS approach according to the distances between the positions of search queries and product locations in the underlying space, and to find out distinctions between viewing behavior (products viewed most frequently) and buying behavior (products purchased most frequently). The results revealed important hints on how to further improve the generation of recommendations.

2 Recommender Systems

A recommender system is software that collects and aggregates information about site visitors (e.g. buying histories, products of interest, hints concerning desired/desirable search dimensions or other FAQ) and their actual navigational and buying behavior and returns recommendations (e.g. based on customer demographics and/or past behavior of the actual visitor and/or user patterns of top sellers with fields of interest similar to those of the actual contact) [3]. Based on such a general description, different target groups for the application of recommender systems can be mentioned, e. g., web site visitors (help with respect to site navigation), online shop operators (support with respect to site structure optimization), or product managers (recommendations with respect to product line design). Taking these considerations as a starting point for our analyses, we were particularly interested in visualizing the lists of available, recommended, viewed and/or purchased products of a retail online shop to support the operation of a recommender system designed to assist site visitors with respect to their buying process. Multidimensional scaling seemed to be an appropriate methodology for this purpose.

3 Property Fitting and Search Query Positioning Within MDS

Multidimensional Scaling (MDS) is the label for a class of methods that represent similarity or dissimilarity values with respect to pairs of objects as distances between corresponding points in a low-dimensional metric space [1]. Graphical display of the object representations as provided by MDS enables to literally "look" at the data and to explore structures visually. A popular and classical technique to construct such object representations is the Kruskal method [4]. The goodness of fit of the solution obtained within the different iterative steps of the method can be assessed by the so called *Kruskal stress*.

One weakness of classical MDS is the difficulty to interpret the object representations in the low-dimensional space. This problem can be overcome by *property fitting*. Let $O = \{o_1, \dots, o_N\}$ be the set of objects and $b_n = (b_{n1}, \dots, b_{nM})$, $n = 1, \dots, N$, the representation of object o_n in the underlying M-dimensional target space. If additional information about the objects is given, e.g., in form of attribute vectors $a_p = (a_{1p}, \dots, a_{Np})'$ for property p , $p = 1, \dots, P$, one can construct property vectors $c_p = (c_{p1}, \dots, c_{pM})$ so that the projections

$$\hat{a}_{np} = \sum_{m=1}^M c_{pm} b_{nm} \quad (1)$$

of b_n onto c_p approximate the actual attribute values of the objects as good as possible with respect to the least squares criterion

$$\sum_{n=1}^N (\hat{a}_{np} - a_{np})^2. \quad (2)$$

Vector notation leads to $c_p = (B'B)^{-1}B'a_p$ with $B = (b_{nm})$. Quality of fit can be measured by correlation coefficients between \hat{a}_{np} and a_{np} .

Similarly we performed the subsequent transformation of the search queries $s_q = (s_{q1}, \dots, s_{q\tilde{P}_q})'$ where the notation search profile is used for a set of at least 10 identical search queries by different web users. Here, $\tilde{p} = 1, \dots, \tilde{P}_q, \tilde{P}_q \leq P$, indicates properties specified in the query. In cases where a range of values was stated, e.g., for the price, we set $s_{q\tilde{p}}$ equal to the mean obtained from the lower and upper boundaries of the specified range. With given property vectors $c_{\tilde{p}}$ we looked for the representation $z_q = (z_{q1}, \dots, z_{qM})'$ of s_q so that the projections

$$\hat{s}_{q\tilde{p}} = \sum_{m=1}^M z_{qm} c_{\tilde{p}m} \quad (3)$$

of z_q onto $c_{\tilde{p}}$ approximate the actual attribute values of the search profiles as good as possible with respect to the least squares criterion

$$\sum_{\tilde{p}=1}^{\tilde{P}_q} (\hat{s}_{q\tilde{p}} - s_{q\tilde{p}})^2. \quad (4)$$

Vector notation leads to $z_q = (C_q' C_q)^{-1} C_q' s_q$ with $C_q = (c_{\tilde{p}m})$.

4 Empirical Analysis

4.1 Prerequisites

For our empirical analysis we used a set of transaction data gathered over a time period from February 5, 2003 until Juli 28, 2003 from a recommender system installed to support the searching behavior with respect to the product class “digital cameras” of website visitors of a large German retail online store. Products were described by price, manufacturer, optical zoom, digital zoom, resolution, memory, manageability and 19 other features which we aggregated to flash, manual control and extras. In this way we ended up with a bunch of ten product properties. Users can formulate a search query by specifying their preferred values in these 10 dimensions. They are then presented a sorted list of products that best match their needs according to an internal, rule-based algorithm. Implicit customer feedback is measured by counting

the number of *product views* which are indicated by clicks on the product image to receive further, more detailed information about the product, and the number of purchase events which are indicated by clicks on the "to-market-basket"-icon. The product catalog consisted of 259 products. Based on the 10 dimensions discussed earlier only 181 different positionings of products remained. Thus, it may happen in a few cases that the location of a frequently viewed or purchased product actually describes more than one product (with differences in those features that we aggregated to flash, manual control or extras). Nearly 70000 search queries appeared in the underlying time period (of which unfortunately about 49000 were empty and may result from web robots). Approximately 80000 product views and 1200 purchase events were recorded. To calculate dissimilarities between products we used the results of the *dist* function from the "R" software package [5]. Multidimensional scaling was done with the help of the function *isoMDS* from the library "MASS" [6] which is an implementation of Kruskal's non-metric MDS [4]. For the solution in the selected two-dimensional space *stress* was 0.167 and still "sufficient". Property fitting was implemented by ourselves as described in section 3. The correlation coefficients for the property vectors were high (> 0.7) for most of the properties. Best fits were achieved for price (0.87) and resolution (0.8), the worst for digital zoom (0.54) and brand (0.47).

4.2 Searching Behavior

As a starting point, we wanted to visualize the products from the product class just described together with the search queries that users mainly specify in the underlying situation. We aggregated identical queries to create so-called search profiles and selected those which represented at least 10 queries. This resulted in 144 search profiles representing 4991 of 22779 search queries (21.9%). These search profiles were mapped into the selected perceptual space as described in section 3. If only one search criteria was specified, we represented this profile as a point on the corresponding property vector instead of depicting a whole subspace (line through that point orthogonal to the considered property vector) since the space representation would have become unreadable otherwise. The result is shown in figure 1.

Two characteristics are remarkable: First, many search profiles are positioned on distinguishable lines. The ones on the negative prolongation of the price vector represent search profiles in which only the desired price was specified. The ones on the other dominant lines result from profiles, where mostly optical zoom and price were specified but only the price varied. Optical zoom was normally set to "two-fold" in these cases. Second, many search profiles are situated in an area which indicates demand for high memory where in contrast only few products are shown. This could be a hint, that customers look for cameras with higher memory than the actually available cameras can offer which could lead to dissatisfaction with the product assortment and reduce willingness to buy. This was confirmed by the observation of product

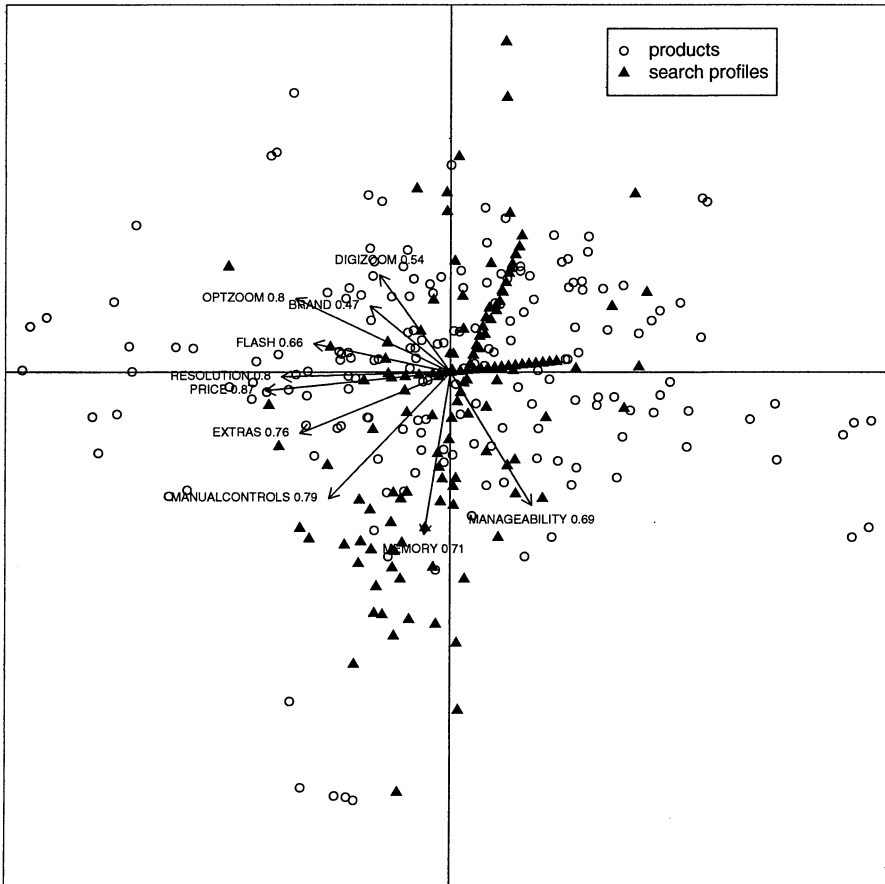


Fig. 1. Representation of search profiles and products

experts that most customers who purchased a digital camera also bought additional memory sticks/cards together with the selected camera.

4.3 Viewing and Buying Behavior

Next, we wanted to examine relations between searching, product viewing and buying behavior of the underlying web visitors. For each search profile we identified the products viewed most frequently and connected graphically their locations with the corresponding search profile position by lines. For three of the most frequent search profiles this is shown in Figure 2. One can see that in general users view products which are positioned relatively close to their specified search profile. However, for all search profiles there are products which are even closer and have not been viewed frequently and others that are relatively far away but had been viewed very often. Comparing

these products with the actual recommendation list - which was not included in our data set - could give some important hints how to further improve the generation of recommender system results.

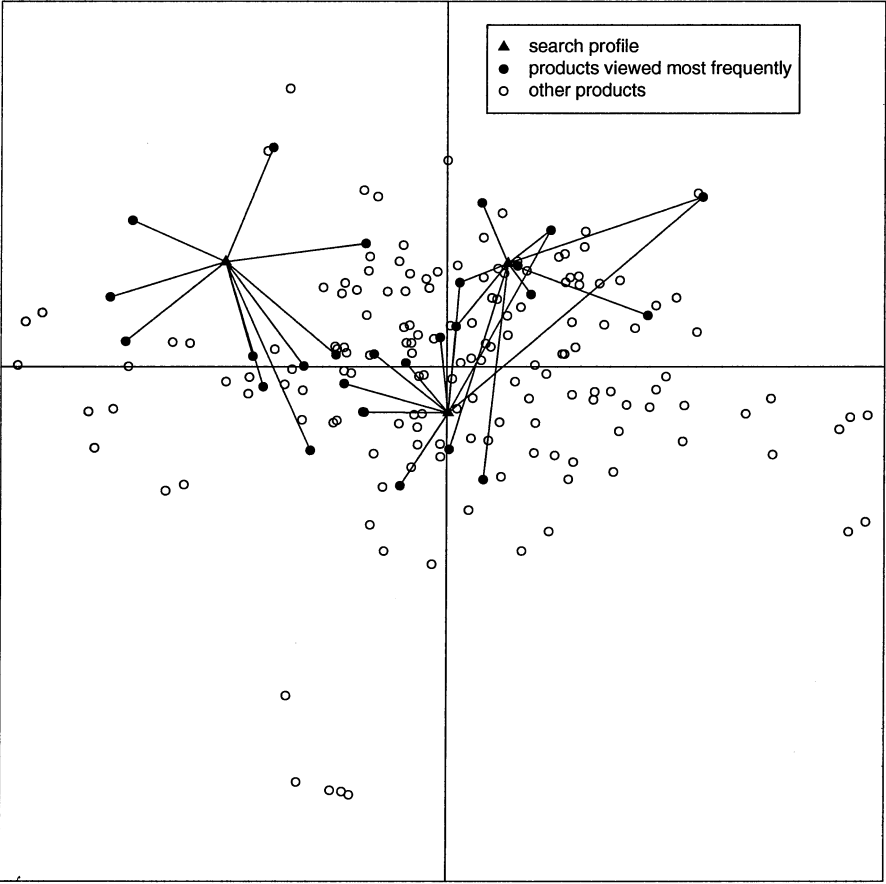


Fig. 2. Selected search profiles and products viewed most frequently

The same is valid for the corresponding analysis for the purchased products. Figure 3 shows that the products most frequently purchased are not necessarily the ones closest to the specified profiles. They might not even be among the most frequently viewed ones. This could, perhaps, be explained by what we sometimes call "Porsche effect": People are very interested in outstanding products but they buy the standard and cheaper alternatives.

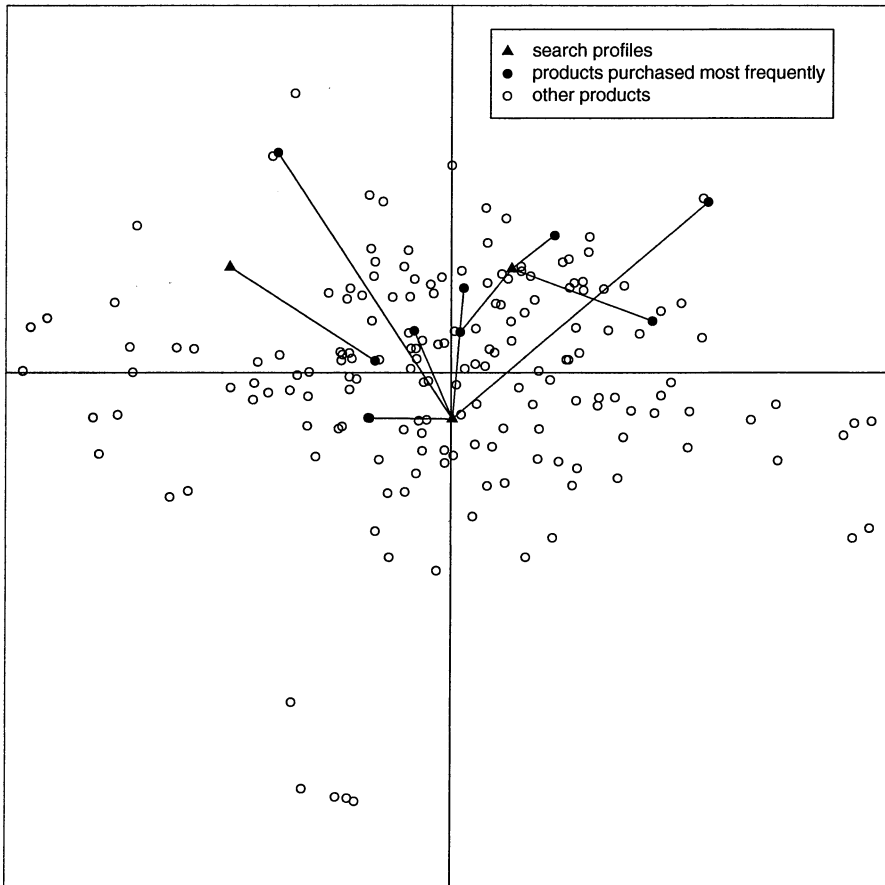


Fig. 3. Selected search profiles and products purchased most frequently

5 Conclusions

In this paper we argued that twodimensional representations of products of a given product class, search queries and recommender system results as well as those products viewed and/or purchased most frequently can be obtained by MDS and displayed together in appropriate spaces to help to visualize the performance of complex recommender systems and the usage of recommender system results by web site visitors. We showed that these visualizations can create valuable insights for product managers as well as for recommender system engineers. We restricted the presentation in different aspects. First, representations are based on distances in the underlying space. Second, only an actual snapshot of the system is represented. Further work will have to address both problems.

References

1. Borg, I., Groenen, P. (1997) Modern Multidimensional Scaling: Theory and Applications, Springer
2. Gaul, W., Geyer-Schulz, A., Schmidt-Thieme, L., Hahsler, M. (2002) eMarketing mittels Recommendersystemen, Marketing ZFP 24, 47-55
3. Gaul, W., Schmidt-Thieme, L. (2002) Recommender Systems Based on User Navigational Behavior in the Internet, Behaviormetrika 29, 1-22
4. Kruskal, J. B. (1964) Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. Psychometrika 29, 1-27
5. The R Development Core Team (20003), The R Environment for Statistical Computing and Graphics,
<http://cran.r-project.org/doc/manuals/fullrefman.pdf>
6. Venables, W.N., Ripley, B.D. (1997) Modern Applied Statistics with S-Plus [3rd ed.], Springer

Die Modellierung von Präferenzveränderungen mittels Scanner Panel Daten

Lutz Hildebrandt und Lea Michaelis

Institut für Marketing, Wirtschaftswissenschaftliche Fakultät, Humboldt-Universität zu Berlin, Spandauer Str. 1, 10178 Berlin, hildebr@wiwi.hu-berlin.de, michaeli@wiwi.hu-berlin.de

Zusammenfassung Kontexteffekte können im Zusammenhang mit Neuprodukt-einführungen eine entscheidende Rolle spielen. Sie wurden bislang primär in Experimenten nachgewiesen. Die vorliegenden Simulationsstudie untersucht die Leistungsfähigkeit eines neuen Ansatzes, der es erlaubt Kontexteffekte in Scanner Panel Daten zu messen.

1. Einleitung

Die erfolgreiche Einführung eines neuen Produktes wird üblicherweise zu einer Veränderung der Markstruktur führen. Eine Erklärung sind die Veränderungen der Präferenzen und Wahrnehmung der Konsumenten. Kontexteffekte können in diesem Zusammenhang eine entscheidende Rolle spielen. Bekannte Beispiele für Kontexteffekte in der Markenwahl sind der Attraction-Effekt [6, 7, 12, 9] und der Compromise-Effekt [11]. Einige Erklärungsansätze führen diese Effekte auf kontextbedingte Wahrnehmungsverzerrungen der Konsumenten, insbesondere auf Range-, Frequency- [12] und Categorization-Effekte [9] zurück. Kontexteffekte wurden primär in experimentellen Studien nachgewiesen. Die meisten experimentellen Studien sind problematisch, da sie auf einem Experimental-/Kontrollgruppen Design basieren und neben der Information über individuelle Vorgänge [14] auch die zeitliche Dimension vernachlässigen. Implizit wird von einer Symmetrie der Effekte des Erscheinens und des Verschwindens von Alternativen ausgegangen [13]. Außerdem betrachten fast alle vorliegenden Studien nur den Fall zweier bestehender und einer eintretenden Marke. Somit können keine Prognosen für Situationen mit mehreren Marken gemacht werden. Studien mit realen Kaufdaten stehen noch immer aus, obwohl sie Aussagen größerer Reichweite liefern können.

Der Mangel an Untersuchungen mit echten Kaufdaten kann auf methodische Probleme zurückgeführt werden. Markenwahlmodelle erscheinen für die Analyse von Kontexteffekten auf den ersten Blick ungeeignet. Die o.g. Präferenzeffekte verstoßen z.B. bei Annahme unveränderlicher Präferenzen gegen die Regularitätsbedingung und die Independence of Irrelevant Alternatives Eigenschaft (IIA-

Eigenschaft), der beispielsweise das Logit-Modell [5] unterliegt. Methodische Weiterentwicklungen können kritische Eigenschaften des Logit-Modells überwinden. So beseitigt die adäquate Berücksichtigung latenter Heterogenität die problematische IIA-Eigenschaft.

In dieser Arbeit wird ein Modell von Chintagunta [4] zur internen Marktstrukturanalyse aufgegriffen, das neben der Erfassung von latenter Heterogenität über einen Finite Mixture (FM) Ansatz, die Schätzung einer Präferenzstruktur im mehrdimensionalen Wahrnehmungsraum erlaubt. Es lässt explizit Korrelationen zwischen den Nutzen von Alternativen zu, die den wahrgenommenen Ähnlichkeiten entsprechen [3]. Mittels einer einfachen Erweiterung des statischen Modells zu einem Modell über zwei Perioden können auf der Ebene individueller Scanner Panel Daten die Eintrittseffekte neuer Marken gemessen werden. Mit Hilfe von Modellrestriktionen lassen sich verschiedene Hypothesen über Kontexteffekte auf die Präferenzstruktur statistisch testen.

Ziel der vorliegenden Studie ist es, die Leistungsfähigkeit des erweiterten Modells in der Messung von Präferenzeffekten zu überprüfen. Eine Simulationsstudie in Anlehnung an die Arbeiten von Andrews, Ansari & Currim [1] und Andrews, Ainslie & Currim [2] (im folgenden als AAC bezeichnet) untersucht die Fähigkeit zur Ermittlung unterschiedlicher Präferenzeffekte in verschiedenen Markteintritts-Szenarien und unter verschiedenen technischen Bedingungen.

2. Die Berücksichtigung des Markteintritts im Finite Mixture Logit-Modell mit Präferenzstruktur

Die bedingte Wahrscheinlichkeit, dass Haushalt i mit gegebenem Parametervektor $\theta_i = \{\beta_{0i}, \beta_i\}$ im Zeitpunkt t die Marke j wählt, lautet

$$P_{ijt} = \frac{\exp(\beta_{0ij} + \beta_i' X_{ijt})}{\sum_{l=1}^J \exp(\beta_{0il} + \beta_i' X_{ilt})} \quad (1)$$

Dabei misst β_i die Response auf die Kovariaten X_{ijt} wie den Marketing Mix und β_{0ij} spiegelt die Präferenz des Haushaltes i für die Alternative j wider. Dieser Term wird als intrinsische Markenpräferenz oder Präferenzkonstante bezeichnet. Das FM-Logit-Modell [8] nimmt an, dass die individuellen Parameter θ_i einer diskreten Heterogenitätsverteilung entstammen, die aus einer finiten Mischung von Parametervektoren und zugehörigen Wahrscheinlichkeiten $\rho(\theta_s)$ gebildet wird, mit

$$\sum_{s=1}^S \rho(\theta_s) = 1$$

Unter der Annahme, dass die intrinsische Markenpräferenz eine lineare Funktion d unbeobachtbarer, markenspezifischer Attribute ist, kann der Präferenzvektor in $\beta_{0i} = A w_i$ zerlegt werden [4]. Die Matrix A der Größe $j \times d$ beinhaltet die Attributausprägungen für jede Marke. w_i sind die Nutzwengewichte, die Haushalt i einem bestimmten Attribut zuweist. In diesem Ausdruck sind nur die Präferenzge-

wichte haushaltsspezifisch, während die Attributausprägungen der Alternativen über alle Haushalte gleich und zeitinvariant sind. Die Nutzegewichte, die Dimensionen des Attributraumes sowie die unbeobachtbaren Ausprägungen der Alternativen auf diesen Dimensionen müssen geschätzt werden. Die konditionale Kaufwahrscheinlichkeit ergibt sich unter Berücksichtigung einer Präferenzstruktur als

$$P_{ijt} = \frac{\exp(a_j w_i + \beta_i' X_{ijt})}{\sum_{l=1}^J \exp(a_l w_i + \beta_i' X_{ilt})} \quad (2)$$

Ändert sich innerhalb des Untersuchungszeitraumes durch das Eintreten einer neuen Marke das Choice Set, kann dies die Präferenzstruktur und damit die Nutzen beeinflussen. Eine Veränderung der Präferenzen für die existierenden Marken bewirkt, dass die Markenwahlwahrscheinlichkeiten nicht nur infolge des größeren Choice Sets proportional sinken, sondern sich auch zueinander verändern. Unter der Annahme unabhängiger Käufe über alle Zeitpunkte, die sich nun über zwei Betrachtungsperioden $\tau=1,2$ mit unterschiedlichen Alternativenmengen ($M_{\tau=1}$ ist eine Teilmenge von $M_{\tau=2}$) erstrecken, kann die Likelihoodfunktion eines Haushaltes i als

$$l(y_i) = \sum_{s=1}^S \left[\prod_{t=1}^{T_{\tau=1}} \left\{ \prod_{j=1}^{J_{\tau=1}} P_{\tau=1,ijt}^{\delta_{ijt}} \right\} \times \prod_{t=1}^{T_{\tau=2}} \left\{ \prod_{j=1}^{J_{\tau=2}} P_{\tau=2,ijt}^{\delta_{ijt}} \right\} \right] \rho(\theta_s) \quad (3)$$

spezifiziert werden. Hier wird vereinfachend angenommen, dass die Wahrscheinlichkeiten der latenten Segmente $\rho(\theta_s)$ gleich bleiben. Die Kaufwahrscheinlichkeit P_{ijt} kann entsprechend (2) modelliert werden. Wir bezeichnen dieses Modell im folgenden als 2PCH. Die Parameter werden simultan für beide Perioden geschätzt. Ausgehend von der restriktivsten Spezifikation, keine Veränderung der Präferenzen für bestehende Marken, können Markteintrittseffekte durch die Freisetzung von Parametern zugelassen werden. Das 2PCH-Modell lässt inhaltlich verschiedene Modifikationen der Präferenzstruktur zu. Modellrestriktionen können sowohl auf die Attribute (Wahrnehmung) als auch auf die Nutzegewichte (Präferenzen) gelegt werden, sie können einzelne Marken und einzelne Segmente betreffen. Somit ermöglicht der 2PCH Ansatz explizit zwischen den verschiedenen Kontexteffekten zu differenzieren. Tabelle 1 gibt einen Überblick über die im Rahmen dieser Studie untersuchten Spezifikationen. Das 2PCH Modell benötigt zur Abbildung des Kaufverhaltens im jeweiligen Szenario deutlich weniger Parameter als das FM-Modell ohne Präferenzstruktur nach (1) (vgl. Tabelle 2).

3. Das Design der Simulation

Der Aufbau der Simulationsstudie ist an der Arbeit von AAC [2] orientiert. Vier technische Faktoren, von denen erwartet wird, dass sie die Leistungsfähigkeit der

Modellbestimmung und Schätzgenauigkeit beeinflussen, werden experimentell manipuliert:

Faktor 1: Anzahl der Marken vor und nach Eintritt: 5/6 oder 8/9

Faktor 2: räumliche Konzentration der Marken: hoch oder gering

Faktor 3: Heterogenität innerhalb der Segmente: $\sigma=0,5$ oder $\sigma=0,25$

Faktor 4: Heterogenität zwischen den Segmenten als Unterschiedlichkeit der Präferenzstrukturen: Präferenzwinkel insgesamt $< 90^\circ$ oder $> 90^\circ$

Nur der Faktor 1 ist in der Realität beobachtbar. Da FM-Modelle keine within-Segment Heterogenität abbilden können, ist Faktor 3 ein Test für die Robustheit der Schätzung bei Verletzung der Annahme von Homogenität innerhalb der Segmente. Darüber hinaus prüft dieser Faktor die Robustheit der unterstellten linearen Präferenzstruktur. Durch die verbleibende Heterogenität innerhalb der latenten Segmente gilt das Präferenzvektormodell für die Konsumenten nur annähernd. Die individuellen Abweichungen sind um so gravierender, je größer die Streuung innerhalb der Segmente ist. Die Konsequenzen mehr oder minder starker Verletzungen auf individueller Ebene werden in den Modellergebnissen sichtbar. Je schwächer die Trennung zwischen den Segmenten, d.h. je höher die Korrelation zwischen den Präferenzkonstanten über die Segmente ist, desto schwieriger ist die Zerlegung in Positionen und Präferenzgewichte. Die Präferenzkonstanten zweier Segmente sind daher um so höher korreliert, je ähnlicher ihre Präferenzvektoren sind. In der Studie wurde die Summe der jeweils kleinsten Winkel zwischen aneinandergrenzenden Vektoren als Kriterium zugrundegelegt und auf zwei Stufen variiert. Die Präferenzvektoren sind relativ ähnlich, wenn sie zusammen einen Winkel kleiner als 90° aufspannen, sie sind relativ unähnlich, wenn sie einen Winkel größer 90° aufspannen. Die Clusterung von Marken im Wahrnehmungsraum, die aus hoher Ähnlichkeit der Präferenzen für verschiedene Marken innerhalb der Segmente resultiert, könnte ebenfalls die Schätzung des Präferenzmodells erschweren. Sowohl die räumliche Konzentration der Marken im Wahrnehmungsraum als auch die Ähnlichkeit der Präferenzen der latenten Segmente reflektieren den Wettbewerb einer Produktkategorie.

Eine Reihe von Faktoren können in dieser Studie nicht manipuliert werden, wenngleich diese in AAC [2] ebenfalls einen Einfluss auf die Güte der Modellschätzung hatten. Um den Umfang der Untersuchung handhabbar zu halten, wurden für diese Faktoren Ausprägungen zugrunde gelegt, die realistischen und durchschnittlichen Ausprägung in empirischen Studien entsprechen. Die Anzahl latenter Segmente beträgt drei, die Stichprobengröße beträgt pro Periode 4500 Beobachtungen und die Mixture Komponenten sind gleichgroß, d.h. jeder Haushalt gehört mit einer Wahrscheinlichkeit von 33,3% zu einer der drei Komponenten. Aus den manipulierten technischen Faktoren ergeben sich somit $2^4=16$ experimentelle Bedingungen, unter denen die Veränderung der Präferenzstruktur untersucht werden soll.

Für jede technische Bedingung werden 5 Markteintrittsszenarien simuliert, die durch die Eintrittsposition der neuen Marke und die Wirkung auf die Präferenz-

struktur gekennzeichnet sind. Sie stellen eine Auswahl der wichtigsten, von der Kontexteffekt-Forschung dokumentierten Effekte dar.

- Szenario 1:** beliebige Eintrittsposition, kein Effekt
- Szenario 2:** dominierter Eintritt gefolgt von Präferenzverschiebung
- Szenario 3:** extremer Eintritt gefolgt von Wahrnehmungsveränderung
- Szenario 4:** extremer Eintritt gefolgt von Präferenzverschiebung
- Szenario 5:** assimilierter Eintritt gefolgt von Wahrnehmungsveränderung

Tabelle 1. Modellrestriktionen zur Repräsentation von Kontexteffekten

	Theoretische Annahme		Modellspezifikation	
			R: restringiert F: frei schätzbar	
Hypothese	A_{entry}	Effekt	A	w
H ₁		Nullmodell	F	F
H ₂	beliebig	kein Effekt	R	R
H _{3, 4, 5}	dominiert	Präferenzeffekt: <i>Attraction-Effekt</i>	R	F [w_s]
H _{3, 4, 5}	extrem	Präferenzeffekt: <i>Compromise-Effekt</i>	R	F [w_s]
H ₆	extrem compromise	Wahrnehmungseffekte: <i>Range-Effekt</i> <i>Frequency-Effekt</i>	F [Alle]	R
H _{7, 8}	extrem assimiliert compromise	Wahrnehmungseffekte: <i>Range-Effekt</i> <i>Categorization-Effekt</i> <i>Frequency-Effekt</i>	V [Set] , R	R

Es ergeben sich insgesamt 80 Datensätze, die jeweils eine pre- und post-entry-Periode sowie eine Validierungsperiode umfassen. Das 2PCH Modell wird für jeden Datensatz auf die acht in Tabelle 2 skizzierten Hypothesen getestet. Da Präferenzeffekte in jedem der drei Segmente auftreten können, werden drei Modelle geschätzt, die auf den Attraction-Effekt (Szenario 2) und den Compromise-Effekt (Szenario 4) testen. Wahrnehmungseffekte können den gesamten Wahrnehmungsraum und somit alle Alternativen betreffen oder sich auf einzelne Marken beschränken. Aus diesem Grund werden für die Wahrnehmungseffekte jeweils drei Modelle geschätzt, die die Menge betroffener Marken sukzessive eingrenzen. Im Null-Modell werden sämtliche Koordinaten und Gewichte der zweiten Periode neu bestimmt, während das restriktivste Modell Gleichheit sämtlicher Koordinaten und Gewichte in beiden Perioden unterstellt. Da der Fokus der Studie auf der Überprüfung verschiedener inhaltliche Hypothesen liegt, wurden die

Gruppengrößen auf den wahren Wert fixiert.¹ In Anbetracht des Rechenaufwandes wird für jede der 80 Bedingungen eine Replikation generiert. Für das gesamte Design sind insgesamt $80 \times 8 = 640$ Modellschätzungen erforderlich.

Jeder Datensatz umfasst 450 Haushalte, die jeweils maximal 10 Käufe in jeder der beiden Betrachtungszeiträume sowie 5 zusätzlich Holdout Käufe in der Post-Entry Periode tätigen. Als exogene Variablen wurden Display, Handzettel (binär) und Preis (kontinuierlich) simuliert. Die binären Variablen wurden per Zufallsziehung so erzeugt, dass sie mit einer Wahrscheinlichkeit von 10% bzw. 15% den Wert Eins annehmen. Die Preisvariable ergibt sich aus der Integration von regulärem Preis und Aktionspreis über eine dritte binäre Variable Promotionaktion mit 15% Wahrscheinlichkeit. Um die Identifikation und Genauigkeit der Parameterschätzung zu gewährleisten, wurden die mittleren Preise so gesetzt, dass für jede Marke, unabhängig, ob diese durch hohe oder niedrige Präferenz gekennzeichnet ist, insgesamt eine ausreichende Zahl von Käufen vorliegt. Im Gegensatz zu AAC [2] wurde zugelassen, dass bestimmte Marken in bestimmten Segmenten nur selten gekauft werden. Die Preisvariablen folgen einer Normalverteilung. Die mittleren segmentspezifischen Parameterwerte, auf die within-Segment Heterogenität über Zufallseffekte addiert wird, betragen -2.0 für den Preiseffekt, 0.9 für den Displayeffekt und 0.4 für den Handzetteleffekt. Die mittleren Präferenzkonstanten wurden so festgelegt, dass sie einem Pre- und Post-Entry Präferenzmodell entsprechen, das einen der fünf simulierten Effekte aufweist. Die Erzeugung der simulierten Kaufdaten entspricht dem üblichen Verfahren [2].

Tabelle 2. Anzahl zu schätzender Präferenzparameter^a

Anzahl Marken	2PCH					2PCH Nullmodell	FM-Logit-Modell nach (1)
	SH_2	$SH_{3,4,5}$	SH_6	SH_7	SH_8	SH_1	
5/6	16	18	23	19/20/21	18/19/20	27	30
8/9	22	24	35	28	26	39	48

^a Präferenzparameter für das 2PCH-Modell sind die Gewichte und Koordinaten abzüglich notwendiger Restriktionen, um Identifizierung zu erreichen.

Das Auffinden des korrekten Modells hängt von der Güte der Schätzung des Präferenzmodells ab. Die Identifikation des Präferenzmodells wird durch die manipulierten Faktoren bedingt. Je ähnlicher segmentübergreifend die Präferenzen für Alternativen sind, desto schwieriger wird es, die wahre Präferenzstruktur zu ermitteln. AAC [2] haben nachgewiesen, dass die individuellen Parameterschätzer im FM-Ansatz bei hoher Heterogenität innerhalb der Segmente, verglichen mit den Schätzern des Hierarchischen Bayes-Ansatzes, überraschend gut sind. Wir erwarten für das 2PCH Modell, dass die Extraktion der Präferenzstruktur und die Identifikation der wahren Szenarien bei hoher Heterogenität innerhalb der Seg-

¹ Für das 2PCH-Modell ist ein schrittweises Vorgehen zur Bestimmung der Anzahl latenter Segmente und guter Startwerte erforderlich. Erst im letzten Schritt folgt der hier betrachtete Hypothesentest über zwei Perioden.

mente schwieriger wird, da die Abweichungen vom unterstellten linearen Präferenzmodell größer werden.

4. Ergebnisse

Tabelle 3 fasst die Ergebnisse für die Trefferquote unter Verwendung alternativer Informationskriterien (Akaike’s information criterion (*AIC*), consistent Akaike’s information criterion (*CAIC*) und Bayesian information criterion (*BIC*)) zusammen.

Tabelle 3. Modellselektion, Anteil richtig identifizierter Szenarien in Prozent

technische Faktoren		BIC	AIC	CAIC	übereinstimmend identifizierte Modelle ^a
<i>Anzahl Marken</i>	5/6	93	88	90	83
	8/9	70	78	68	48
<i>Konzentration</i>	hoch	73	78	70	55
	gering	90	88	88	75
<i>Heterogenität within</i>	hoch	83	80	80	65
	gering	80	85	78	65
<i>Heterogenität between</i>	hoch	78	85	75	63
	gering	85	80	83	68
Gesamt		81	83	79	65

^a Ein Modell ist übereinstimmend als richtig identifiziert, wenn es durch alle drei Informationskriterien ausgewählt wird.

1. Den größten Einfluss auf die Leistungsfähigkeit des Ansatzes zur Identifikation alternativer Präferenz- und Wahrnehmungseffekte haben die Faktoren *Anzahl der Marken* und *räumliche Konzentration*. Im 5/6-Marken-Fall gelingt es besser, das korrekte Szenario zu ermitteln. Der Effekt sinkender Beobachtungen pro Marke dominiert offensichtlich den relativen Zuwachs an Freiheitsgraden für das Präferenzmodell, verglichen mit dem FM-Modell ohne Präferenzstruktur.

2. Die Ähnlichkeit der Präferenzen, die sowohl über den Faktor *räumliche Konzentration* als auch den Faktor *Heterogenität between* manipuliert wurde, führt nur im ersten Fall zu signifikanten Unterschieden in der Trefferquote, während sich die Trennung der Mixture Komponenten in dem hier variierten Ausmaß nicht signifikant auswirkt.

3. Die *Heterogenität innerhalb der Segmente* hat keinen signifikanten Einfluss auf die Treffergenauigkeit. Der 2PCH Ansatz erweist sich als relativ robust gegenüber der daraus resultierenden Verletzung des unterstellten Präferenzmodells. Dieses Ergebnis steht im Einklang mit den Erkenntnissen von AAC [2], die dem FM-Ansatz eine hohe Robustheit gegen die Verletzung der Annahme von Homogenität innerhalb der Segmente nachgewiesen haben.

Insgesamt erzielt der vorgeschlagene Ansatz gute Ergebnisse bei der Aufdeckung des korrekten Szenarios. Unter Verwendung verschiedener Informationskriterien konnte der korrekte Markteintrittseffekt in 65% der Fälle sogar übereinstimmend identifiziert werden, während die a priori Wahrscheinlichkeit jedes Modells in dieser Studie nur 1/8 beträgt. Im Zweifelsfall kann zusätzlich der Likelihood-Ratio-Test herangezogen werden, da einzelne Hypothesen ineinander genistet sind. In vielen Fällen konnten Unklarheiten, die aus unterschiedlichen Ergebnissen der Informationskriterien resultieren, beseitigt werden, so dass die Trefferquote entsprechend höher liegt.

Literatur

1. Andrews RL, Ansari A, Currim IS (2002) Hierarchical Bayes Versus Finite Mixture Conjoint Analysis Models: A Comparison of Fit, Prediction, and Partworth Recovery, *Journal of Marketing Research*, 39, 87-98.
2. Andrews RL, Ainslie A, Currim IS (2002) An Empirical Comparison of Logit Choice Models with Discrete Versus Continuous Representations of Heterogeneity, *Journal of Marketing Research*, 39, 479-487.
3. Brownstone D, Train K (1999) Forecasting New Product Penetration with Flexible Substitution Patterns, *Journal of Econometrics*, 89, 109-129.
4. Chintagunta PK (1994) Heterogeneous Logit Model Implications for Brand Positioning, *Journal of Marketing Research*, 31(2), 304-311.
5. Guadagni P, Little J (1983) A Logit Model of Brand Choice Calibrated on Scanner Data, *Marketing Science*, 2, 203-238.
6. Huber J, Payne JW, Puto C (1982) Adding Asymmetrical Dominated Alternatives: Violations of Regularity and the Similarity Hypothesis, *Journal of Consumer Research*, 9, 90-98.
7. Huber J, Puto C (1983) Market Boundaries and Product Choice: Illustrating Attraction and Substitution Effects, *Journal of Consumer Research*, 10, 31-43.
8. Jain DC, Vilcassim NJ, Chintagunta PK (1994) A Random-Coefficients Logit Brand-Choice Model Applied to Panel Data, *Journal of Business and Economic Statistics*, 12, 317-328.
9. Pan Y, Lehmann DR (1993) The Influence of New Brand Entry on Subjective Brand Judgements, *Journal of Consumer Research*, 20, 76-86.
10. Parducci A (1974) Contextual Effects: A Range-Frequency Analysis, in: Handbook of Perception, Vol. II, eds. Carterette EC, Friedman MP, New York: Academic Press.
11. Simonson I. (1989) Choice Based on Reasons: The Case of Attraction and Compromise Effects, *Journal of Consumer Research*, 16, 158-174.
12. Simonson I, Tversky A (1992) Choice in Context: Tradeoff Contrast and Extremeness Aversion, *Journal of Marketing Research*, 29, 281-295.
13. Sivakumar K, Cherian J (1995) Role of Product Entry and Exit on the Attraction Effect, *Marketing Letters*, 6, 45-51.
14. Stewart, DW (1989) On the Meaningfulness of Sensory Attributes: Further Evidence on the Attraction Effect, *Advances in Consumer Research*, 16, 197-202.

Measurement of Online Visibility

Nadine Schmidt-Mänz and Wolfgang Gaul

Institute of Decision Theory and Management Science, University of Karlsruhe,
Germany

Abstract. To attract web visitors via the internet it is fundamental for all kinds of online activities to be "visible" in the net. Visibility measurement is important for web sites: it helps to define benchmarks with respect to competition and allows to calculate visibility indices as predictors for site traffic.

This paper discusses a new approach to measure online visibility (OV) and compares it with one known from the literature.

We describe physical and psychological drivers of OV and suggest a measurement of OV that works automatically as a robot in the internet. Managerial implications to make web sites "smelly" or "visible" are also discussed.

1 Introduction

Nearly 80% of all internet users find new web sites with the aid of search engines. With this information in mind, search engines appear to be very important instruments to get in contact with new customers for whatever online business or visitors of web sites in this medium. For example, an insurance broker listed at the first position of the result page of a search engine may expect to be frequented by up to 10,000 visitors per month when web users look for "insurance" [6]. A traditional marketing campaign would cost several thousand dollars to have the same effect. Hence, it is very important to "maximize" what could be called online visibility (OV) of web sites, or -at least- to improve OV relative to competition.

In section 2 we describe main drivers of OV and focus on facts about human online searching behavior. This leads to a new measure of OV and possibilities for its determination as discussed in section 3. In section 4 we give some managerial implications. Section 5 presents conclusions.

2 Drivers of Online Visibility

OV is composed of different parts. Here, we will concentrate on the following two: There are psychological drivers of OV derived by human online searching behavior. Here, questions like how humans use the internet and how they interact with search engines have to be taken into consideration. Psychological drivers can decrease OV. Additionally, physical drivers of OV such as links to a web site, banner ads, listings in search engines or directories, etc., are of importance. They can be influenced by administrators of web sites themselves and increase OV.

2.1 Psychological Drivers of Online Visibility

To measure OV it is essential to understand psychological drivers derived by human online searching behavior:

As already mentioned about 80 % of all internet users find new web sites with the aid of search engines. Thus, *Google*, *Teoma*, etc., are effective instruments to reach new visitors or customers [6].

In course of time, people become more efficient in using the web by navigating directly to a web site they already know, but they use search engines to find new ones (see table 1 according to an independent report by *WebSideStory*, Inc. [14]).

Table 1. Global Internet Usage

Referral Type	2002	2003	trend
Direct Navigation	50.21%	65.48%	↗
Web Links	42.60%	21.04%	↘
Search Engines	07.18%	13.46%	↗

As global internet usage with respect to search engines has nearly doubled from 2002 to 2003 the question is how do internet users interact with search engines? There are three main studies covering web searching strategies with the help of search engines by analyzing query logs: The *Fireball* [8], the *Excite* [9], and the *AltaVista* study [12].

In the following we only concentrate on the *AltaVista* study, since conclusions of all three studies are nearly the same. [12] is based on the largest data set (one billion queries submitted to the main *AltaVista* search engine over a 42-day period) and provides a broad spectrum of information: given number of queries in the data set, length of the collection period and analysis, it is the most complete web searching study to date.

Among the facts worth mentioning about human online searching behavior the following are interesting:

Nearly 77.6% of all query sessions consisted of only one request. 85.2% of searchers examined only one result screen per query (7.5% two, and 3.0% three screens). The average number of terms in a query added up to 2.35 ($\sigma = 1.74$) and that of operators (AND, OR, NOT ...) to 0.41 ($\sigma = 1.11$). According to the total number of queries, 63.7% occurred only once, i. e. the formulated queries were "almost" unique. The most popular term in a query was "sex" with an appearance of 1,551,477 times. This is equal to 2.7% of the total number of non-empty queries.

2.2 Physical Drivers of Online Visibility

Drèze and Zufryden [5] defined visibility as the extent of presence of a brand or a product in the consumer's environment. Thus, one can view OV as an indicator with respect to potential web site traffic in the same way as awareness is a precursor to purchase. A web site can draw attention to its content and attract potential internet surfers by both offline (television, radio, newspapers, etc.) and online (banner ads, links from search engine result pages, online directories or other web pages, etc.) means.

The authors identified three main physical drivers which increase OV and can be measured by taking a snapshot of the internet.

Links from other web sites: One can increase OV by increasing the number of incoming links from important or frequently visited web pages. To find the amount of incoming links from web pages to the underlying site one can use the brute force approach, i. e. crawl the net and keep track of all links located on web sites. This is the common approach of most search engines which provide users with a list of these links (fan-ins, inbounds, incoming links). Users can retrieve this kind of information by a special command and the corresponding URL in the search interface (e.g., "link:www.xyz.com" with respect to *Google*) and should repeat this procedure for different engines to ensure maximum coverage [2].

There is a serial position effect in the HTML code that relates to effectiveness of links (the higher the rank order of a link the higher the click through rate on that link [1]) The hypothesis that the effectiveness of links is also affected by the depth of the page on which the link was located (with the home page of a web site having a depth of 1) was statistically not significant.

Online Directories: To become visible in the net it is important to be listed in the adequate categories of online directories. Directories or subject catalogs that represent the "yellow pages" of the internet are characterized by their hierarchical structure. The most important property of directories is that they are validated by human beings and not by robots. Experts look at every page submitted for registration in the index and review it. This procedure is generally referred to as *manual indexing*. One advantage is that a human indexer can detect relations between seemingly different contents. But it is very time consuming and expensive. Well known examples for directories are *Yahoo!* [15] and *ODP* [13].

Search Engines: Web search engines (also called web indices, index servers, or simply search engines) work exclusively with automated methods [11]. A collection of high rankings in such engines is one of the best prerequisites to yield enhanced OV for new visitors. Drèze and Zufryden [5] took the fact into consideration that search engines always return result pages which are formatted in a special way (e.g. 10 links per result page) and that the frequency

of looking at the first, second, third ... result page is decreasing.

In addition, they characterized web sites as trees that surfers explore by going up and down branches where less weight was assigned to deep links and more to links that are close to the root of the respective site.

These results gave some insights in how people process online information. First, the position of a link on a web page is important for OV. Second, as it has not been proved that people view web sites structured in an arborescent manner, the depth of a page in the site seems to have little impact on its importance. This may be due to the ability of search engines to route people directly to that page where the desired information is located. Additionally, it should be noted that, in the meantime, it is legalized that search engines use these "deep links" [4]. Third, the results show that visibility of web sites in search engines is dependent on both position of the link on and depth of the result page on which the underlying link is situated.

3 A New Approach to Measure OV

Based on the knowledge of human online searching behavior and the functioning of ranking algorithms of search engines such as PageRank [3,7,10] we can define several (new) impacts on OV.

3.1 Impacts of Human (Online) Searching Behavior on OV

One aspect to define a new measure of OV is the importance of fan-ins and directory listings on ranking algorithms of search engines. Another aspect is related to the searching behavior of internet users. If OV is measured by the three physical drivers mentioned before (search engine ranking, incoming links, and directory listings) in the same way, some problems may arise. To calculate a precise measurement one will have to subtract all overlappings and as such a measure doesn't exist we have estimated OV based on what people will definitively see.

1. The first search engine results visible on the answer screen as a cutout of the first result page (in the majority of cases ten entries) will have the best chance to be clicked on by a searching person
2. Human searchers don't browse every result page of a search engine. They only browse a few with exponentially decreasing intensity (*AltaVista* study [12]).
3. We do not incorporate the number of results returned by search engines, as most search engines return ten entries per result page.
4. In addition, we consider a measurement of Adwords' appearance (a special feature of *Google*) because this is a good instrument for enterprises to become visible via up-to-date activities (special offers, sales) in a short time with the aid of bundles of keywords.

5. We do not use general keywords from keyword databases but propose an individualized set of keywords to differentiate web sites by content.
6. We do not only measure OV with one set of keywords but use a set of up to three keywords and all subsets of variations.
7. As it is not possible to measure OV in directories in an objective way (the best-known directories list URLs or names in alphabetical order like yellow pages) we omit the determination of visibility via listings in online directories.
8. We count fan-ins to a certain extent: we include incoming links with a small factor in order not to overestimate this effect.

3.2 The New Measure *GOVis*

Our new approach to estimate OV has been called GOVis (Gage of Online Visibility) to differentiate it from other measures. It is given by

$$GOVis = \sum_{k=1}^N \sum_{n=1}^{\binom{N}{n} \cdot n!} \left[\alpha \cdot \sum_{p=1}^2 \frac{1}{e^{p-1}} \cdot \sum_{r=1}^R X_{kpr} + \beta \cdot \sum_{a=1}^A Y_{k1a} \right] + \gamma \cdot \ln(Z_L) \quad (1)$$

with

- * \mathcal{K} is a set of interesting keywords, with $|\mathcal{K}| = N$ (normally $N \leq 3$), $\sum_{n=1}^N \binom{N}{n} \cdot n!$ is the quantity of all ordered subsets of $\wp(\mathcal{K}) \setminus \{\emptyset\}$ and k is the kth subset of keywords with which a query in *Google* is performed,
- * p is the depth of the result pages of the search engine used (we have restricted depth to $p \leq 2$),
- * R is the quantity of results per result page and r is the rth ranking position on the result pages (we, normally, use *Google* with $R = 10$),
- * A is the maximum quantity of Adwords per result page (*Google* standard is $A = 8$) and a is the ath Adword ranking position,
- * L is the corresponding URL, Z_L is the number of corresponding fan-ins,
- * h_{kpr} is the hyperlink at the rth ranking position on the page with depth p generated by a query with the kth subset of keywords,
- * w_{k1a} is the Adword link at the ath Adword ranking position on the page with depth 1 generated by a query with the kth subset of keywords,
- * $X_{kpr} = \begin{cases} 1, & h_{kpr} \text{ links to } L \\ 0, & \text{otherwise} \end{cases} \quad Y_{k1a} = \begin{cases} 1, & w_{k1a} \text{ links to } L \\ 0, & \text{otherwise} \end{cases}$
- * $\alpha + \beta + \gamma = 1$.

In (1) we excluded measures of OV in directories, on portals, in chat rooms or banner ads, etc. We didn't consider overlappings, but we used different numerical values for α, β , and γ to consider how GOVis depends on differences with respect to the meaning of fan-ins, Adwords or rankings in search engine result pages.

3.3 Implementation and Results of *GOVis*

We implemented (1) in PHP as a parser. We used *Google* and Adwords for queries (because *Google* has the most pages in the index). For the quantity of fan-ins we considered *AltaVista* (as in *Altavista* it is possible to exclude links from the home domain (link:http://www.xyz.com -host:http://www.xyz.com) while *Google* only saves fan-ins of pages with a high PageRank). In this first version of *GOVis* it is possible to enter up to three keywords, a special URL and to customize α , β , and γ . For results of OV measured by *GOVis* we used scenarios with different α , β , and γ for $\mathcal{K}=\{\text{Bücher, Dvds, HiFi}\}$, see table 2.

Table 2. *GOVis* results

scenario _x (α , β , γ)	enterprise _x	$\sum X_{kpr}$	$\sum Y_{k1a}$	$\ln(Z_L)$	<i>GOVis</i>
0.15, 0.00, 0.85	<i>Amazon</i>₁	3+1	4	14.1774	12.5560
	<i>Ebay</i> ₁	0+6	1	11.7464	10.3155
0.40, 0.00, 0.60	<i>Amazon</i>₂	3+1	4	14.1774	9.8536
	<i>Ebay</i> ₂	0+6	1	11.7464	8.3722
0.85, 0.00, 0.15	<i>Amazon</i>₃	3+1	4	14.1774	4.9893
	<i>Ebay</i> ₃	0+6	1	11.7464	3.6381
0.80, 0.15, 0.05	<i>Amazon</i>₄	3+1	4	14.1774	4.0032
	<i>Ebay</i> ₄	0+6	1	11.7464	2.5030

In scenario₁ and scenario₂ we chose α and γ on the basis of adjusted figures from table 1 of the years 2002 and 2003 without "Direct Navigation". The number of users that find new web sites with the aid of search engines increases while on the other hand fewer users reach new pages over web links. In scenario₃ we defined α and γ according to PageRank. Here, γ is the probability that a random surfer continues to click on links while with $1 - \gamma$ (s)he will jump to any page in the web. In this scenario search engines are the best possibility to jump to any visible page in the web (where we have the information in mind that 80% of all internet users find new web pages with the aid of search engines). In scenario₄ we also included the visibility of Adwords to a small extent and lowered the impact of incoming links on ranking algorithms. As it is important to know more about the navigational behavior of target segments to which the efforts to increase OV are directed the calculation of α , β and γ is essential for managerial implications.

3.4 Improvements by *GOVis*

First of all, *GOVis* can be individualized, i. e. that for every company and their competitors OV is measured by content specific keywords. As *GOVis* does not use a general list of most common used keywords one can select keywords depending on special up-to-date activities (sales, new products). This and the fact that we base visibility measurement on human searching behavior (only a few keywords for the formulation of a query, only a few result pages of the search engine are actually considered) lead to short running times to compute OV via *GOVis*.

GOVis is also an effective instrument to control visibility of online activities in relation to competitors. It represents a relative measure of OV based on special keywords concerning the content of and activities related to the corresponding web site.

Since α , β , and γ are customizable one can calibrate these parameters on the basis of new values from new online surveys concerning human searching behavior and the way how target segments interact with the internet to find web sites.

4 Managerial Implications

The scenarios in table 2 show that it becomes more and more important to increase search engine visibility, but that it is also important to take into consideration the searching person itself. Thus, it is not the ultimate to optimize a web site for better search engine ranking. In the end, this can be a tit-for-tat strategy because search engines will adapt their ranking algorithms to block attempts that only optimize web sites with respect to a high ranking position in their result pages. An honest way (with customers/visitors and search engines in mind) to improve OV or to make web sites "smelly" is to observe the log files of the corresponding web site to detect search engine referrals that include important keywords of searching persons. Another possibility is to sift online keyword databases to compare already used keywords with descriptions or text content of the corresponding web site to meet customer needs with respect to, e.g., content, special topics or product descriptions. In case of up-to-date activities it is advisable to buy bundles of Adwords as search engines crawl web sites only every few months, so temporal activities won't be visible.

5 Conclusion

In total the measurement of OV should represent a benchmark for specialized content or activities and not lead to a strategy to become listed at the first position of a search engine result page based on a query corresponding to

keywords describing a general topic. There is an impact of the order of keywords in a query on visibility, thus, long term optimization of content of the corresponding web site has to be considered. Visibility has to be measured in constant short term periods as the internet changes "quickly".

The bundle of content visibility, Adwords visibility, search engine visibility, and visibility based on incoming links describes the most important instruments to account for OV. GOVis is a first step in order to measure online visibility affects.

References

1. Ansari, Asim and Mela, Carl F. (2002) e-Customization. Forthcoming in Journal of Marketing Research, Preprint: <http://www.cebiz.org/downloads/ecustomization.pdf>.
2. Bradlow, Eric T. and Schmittlein, David C. (2000) The Little Engines That Could: Modeling the Performance of World Wide Web Search Engines. Marketing Science, Vol. 19, No. 1, 43-62.
3. Brin, Sergey and Page, Lawrence (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. Proceedings of the 7th International World Wide Web Conference (WWW7), 107-117.
4. Bundesgerichtshof, Mitteilung der Pressestelle (Nr.96/2003) Internet-Suchdienst für Presseartikel nicht rechtswidrig. Urteil vom 17. Juli 2003 - I ZR 259/00.
5. Drèze, Xavier and Zufryden, Fred (2003) The Measurement of Online Visibility and its Impact on Internet Traffic. To appear in: Journal of Interactive Marketing, Preprint: <http://www.xdreze.org/Publications/Visibility.html>.
6. Fischerländer, Stefan (2003) Websites Google-gerecht gestalten - Ganz nach oben. iX 08/2003, 84-87.
7. Henzinger, Monika, Heydon, Allan, Mitzenmacher, Michael, and Najork, Marc (1999) Measuring Index Quality Using Random Walks on the Web. Proceedings of the 8th International World Wide Web Conference (WWW8), 213-225.
8. Hölscher, Christoph (1998) How Internet Experts Search for Information on the Web. In: H. Maurer & R. G. Olson (Eds.), Proceedings of WebNet98 - World Conference of the WWW, Internet & Intranet. Charlottesville, VA: AACE.
9. Jansen, Bernard, Spink, Amanda, and Saracevic, Tefko (2000) Real Life, Real Users, and Real Needs: A Study Analysis of User Queries on the Web. Information Processing and Management, 36: 207-227.
10. Kleinberg, Jon (1998) Authoritative Sources in a Hyperlinked Environment. In: Journal of the ACM, Vol. 46(5), 1999, 604-632.
11. Michael, W. Berry and Browne, Murray (1999) Understanding Search Engines - Mathematical Modeling and Text Retrieval. Software, Environments, and Tools, SIAM, Society for Industrial and Applied Mathematics, Philadelphia.
12. Silverstein, Craig, Henzinger, Monika, Marais, Hannes, and Moricz, Michael (1999) Analysis of a Very Large AltaVista Query Log. *SIGIR Forum*, 33(1): 599-621.
13. www.dmoz.com, last visited August 10, 2002
14. www.websidestory.com. Search Engine Referrals Nearly Double Worldwide, According to WebSideStory, March 2003, last visited August 10, 2003.
15. www.yahoo.com, last visited August 10, 2003.

Determinants and Behavioral Consequences of Customer Loyalty and Dependence in Online Brokerage - Results from a Causal Analysis

Staack, Yvonne

Ph.D. student at Etufo-Institute (Entscheidungstheorie & Unternehmensforschung, Prof. W. Gaul), University of Karlsruhe (TH); E-Mail: ystaack@web.de

Abstract

With the rise of the internet, online brokerage emerged as a fast and cost effective way to trade securities. New competitors naturally focused on generating traffic to acquire many customers. However, in ebusiness, "natural" exit barriers (e.g., location) lose importance. As competitors are only a mouse-click away, online firms must convert first-time users into regular customers to regain acquisition costs. But what determines a customer to be committed to his online broker? And, if he can be successfully retained, what effect will that have on the company's profits? Based on theoretical research, a generic model for the development, dimensions, and behavioral consequences of customer retention are proposed. Using causal analysis, this loyalty/dependence-model is successfully tested in online brokerage.

1 Introduction

The management goal of "customer loyalty" [8] is typically justified by its proposed advantages for the suppliers' profits: repeat purchases, cross-buying, selection of higher priced offerings, recommendations to friends, higher tolerance towards mistakes [9, 10, 23]. Although these theoretical implications seem intuitive, they could only partly be proven empirically. In addition, newly established online businesses have not received enough attention in marketing research.

The following study thus has two main goals: First, the proposed behavioral consequences of "customer retention" will be studied within the context of an internet-based service provider (an online broker). Additionally, it will be analyzed what determines the development of this desired customer status. Second, the study also aims to find out if the distinction of two separate commitment dimensions is practically relevant (e.g., causes different customer behavior, or are influenced by diverse determinants). With respect to the outlined research goals, the paper consists of two main parts: Chapter 2 contains the theoretical foundations which lead to the proposed hypotheses for the empirical test. In chapter 3, the research design is outlined and the results of the causal analysis will be discussed.

2 Theoretical Foundations

2.1 Dimensions and Determinants of Customer Commitment

Most research conceptualize the commitment construct by its (assumed) behavioral consequences [20, 23]. As these effects are to be analyzed, a different approach is needed. Consistent with exploratory research of EGGERT, customer commitment in this study is defined as an "inner state of the customer" towards a specific supplier [11] and operationalized as a two-dimensional construct: The *affective* component of customer commitment (*loyalty*) originates from a sense of attachment to the supplier. This appreciation not only indicates higher repurchase intent, but also resistance to counter-persuasion and a certain willingness to recommend the provider to others [9]. The *cognitive* commitment dimension (spurious loyalty, *dependence*) includes the rather involuntary binding of the customer due to constraints which hamper his switching to another supplier [11]. Due to the involuntary nature of the dependence perception, a potential drawback effect ("reactance") can be expected towards the level of attitudinal loyalty, resulting in a negative causal hypothesis from "dependence" to "loyalty".

Research within economic (i.e., resource-dependence view, transaction cost theory) and customer behavior concepts (i.e., dissonance, learning, risk, social interaction theory) revealed 5 supposed determinants for customer commitment. The first potential cause for high customer commitment is the overall level of *satisfaction* which is defined as the result of a cognitive and affective evaluation process, during which the customer compares the perceived performance of a supplier with its expected or desired performance. The resulting satisfaction level therefore refers to the aggregate experiences with this provider [15]. Second, it is proposed that customer relationships endure because buyers face significant transaction costs or other *barriers to switch* to another supplier. "Switching costs" include financial consequences, but also time, psychological effort, or social constraints connected with the migration [16]. Based on this definition, positive influences from 3 distinct types of migration barriers - *economical, psychological, and social barriers to switch* - on the level of customer commitment are postulated within the research model. Finally, a presumably negative effect on the level of customer commitment is postulated to originate from the customers' perception of the *variety/attractiveness of alternative suppliers* available. It is also proposed that attractive competitive offerings negatively influence a consumer's satisfaction with the current provider and thus indirectly reduce his loyalty.

2.2 Behavioral Consequences of Customer Commitment

The effects of higher customer retention can be explained by dividing "profit" into its factors "number of customers" and "profit per customer". As the total number of customers depends on the retention of current users and the acquisition of new clients; increased customer commitment is proposed to have positive effects on

customer retention ("*continuation of the relationship*") and their intention to communicate positively about their preferred broker ("*positive word-of-mouth/referrals*") which is a credible and cost-effective means of gaining new customers [22].

On the other hand, the average profitability per customer typically improves during the business relationship [26]; a pattern found to be even more extreme in internet businesses [25]. Since profits rise with increasing revenues or falling costs per client, they can be attributed to 3 factors: First, as loyal customers tend to consolidate demand at few suppliers, average purchase amounts are likely to rise over time and cross-/up-buying might occur ("*expansion of the customer relationship*", [13]). Second, as committed clients are more likely to be in a habitual purchasing mode, they are less likely to take advantage of competitors' offers [10]. Thus, lower "*price sensitivity*" is proposed for long-term customers who, on average, pay a price premium [5]. Third, increased customer profits might also be caused by reduced "*customer-specific service costs*" of ongoing business relationships [1, 26].

In total, 5 behavioral consequences of customer commitment were identified from existing research. While the "loyalty"-dimension is proposed to exert only positive influences on each of the behavioral intentions, "dependence" is assumed to have no significant effect on "referrals/word-of-mouth" and negative effects on the "expansion of the relationship" or "service costs". Beyond the direct influences of customer commitment, the intention to "expand the relationship" will be positively influenced by the decision to "continue the relationship".

3 Empirical Results

To test the derived hypotheses, structural equation modeling was chosen because of its ability to simultaneously test/quantify causal relationships between model-exogenous/endogenous latent variables and explicitly integrate measurement error into the analysis [2, 14, 17] (for a description of causal modeling refer to [3, 7]). Primary data for the study was collected by an online survey of 60,000 online brokerage users in July 2002. With 4.591 complete questionnaires, the required sample size (at least 5 times the number of estimated parameters, [4]) was reached. Scales for latent constructs consisted of at least 3 indicators [6]. Variables were measured on 6-step, symmetric Likert scales with extreme points of "completely (dis)agree" or "very (un)likely". AMOS 4.0 was used for parameter estimation.

Measurement models of all latent variables were tested as for reliability and validity: Each construct was required to show a Cronbach's Alpha (measure of internal consistency of all indicators measuring the same factor [18, 24]) of at least 0.7. Based on exploratory factor analysis (principal axis factoring, oblique rotation), all scale items needed to show sufficient loadings on their proposed factor and explain at least 50% of that factor's variability [18]. Confirmatory factor analysis required each measurement model to fulfill global/local fit criteria for causal modeling (i.e., GFI > 0,9; AGFI > 0,9; RMR < 0,1; NFI > 0,9; individual item reliability > 0,3; factor reliability > 0,6; average variance extracted > 50% [3, 4, 19]).

3.1 Determinants of Online Customer Commitment

Five slightly different causal structures fulfilled the outlined fit criteria. The best-fitting partial model for the determinants of customer loyalty and dependence in online brokerage was selected based on best overall fit achieved with as few estimated parameters as possible. Since all models were "nested", the chi-square-difference-test was applied to compare the fit of the causal structures [8, 21]. Figure 1 contains the best-fitting partial model of the commitment determinants.

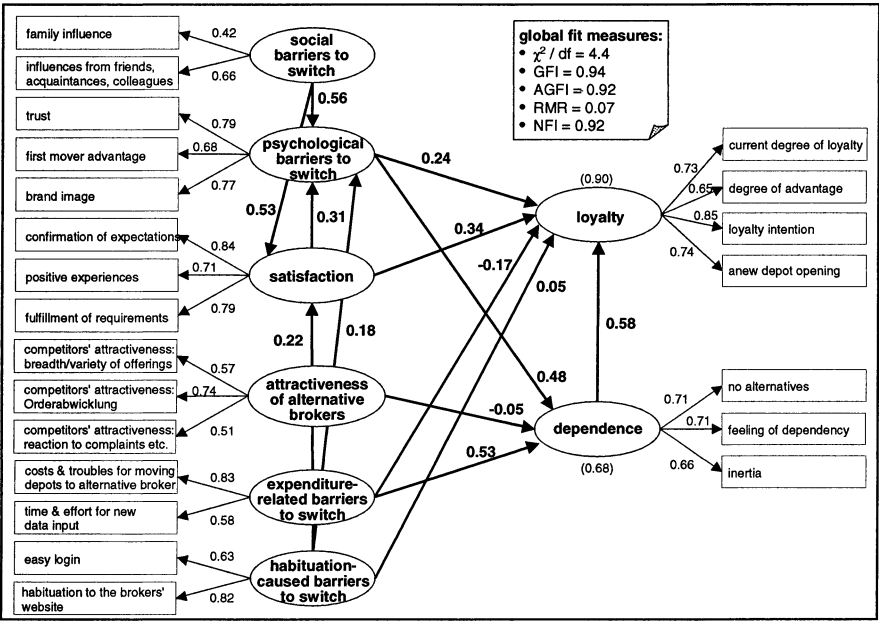


Fig. 1. Best-fitting partial model for the determinants of customer commitment

All global fit measures of this causal model meet or exceed their target levels. In total, the causal structure shows a high explanatory power since 90% of the target construct "loyalty" and 68% of the "dependence"-dimension can be explained by the included 6 determinants¹ of the two-dimensional customer commitment construct. As for the tested relationships, 10 show the postulated sign and statistical significance at the 5% level. The supposedly negative path coefficient from "dependence" to "loyalty" turns out to be highly significant but with a strong positive sign. An explanation for this phenomenon can be found in dissonance theory: As an individual seeks to reduce the dissonances that originate from tensions between the persons attitudes and his behaviors, the possibly low level of affective "loyalty" might in retrospect be increased if the person perceives a high degree of "dependence" from his current supplier [12]. By closing the gap between both

¹ Based on measurement model validation, "economic barriers to switch" were split into 2 factors: "expenditure-related barriers" and "habituation-caused barriers to switch".

commitment dimensions, higher levels of perceived dependence may then translate into higher declared loyalty towards the provider.

As some determinants influence the target constructs directly (for example, psychological barriers to switch → loyalty), while others affect the endogenous variables indirectly (e.g., social barriers to switch → satisfaction → loyalty). To quantify the total influence of each determinant towards the customer commitment dimensions, direct and indirect effects need to be added (as shown in table 1).

determinant	target construct	direct effect	indirect effect(s)	total effect
satisfaction	loyalty	0.34	0.16	0.50
social barriers to switch		--	0.56	0.56
psychological barriers to switch		0.24	0.28	0.52
expenditure-related barriers to switch		-0.17	0.31	0.14
habituation-caused barriers to switch		0.05	0.20	0.25
attractiveness of alternative brokers		--	- 0.03	-0.03
dependence		0.58	--	0.58
satisfaction	dependence	--	0.15	0.15
social barriers to switch		--	0.35	0.35
psychological barriers to switch		0.48	--	0.48
expenditure-related barriers to switch		0.53	--	0.53
habituation-caused barriers to switch		--	0.12	0.12
attractiveness of alternative brokers		-0.05	--	-0.05

Table 1. Total effects of the determinants of customer commitment

Main determinants of the loyalty of online brokerage users are social and psychological switching barriers (i.e. trust; total effect = +0.56 and +0.52, respectively) and satisfaction with their current experiences (+0.50). As for perceived dependence, economic barriers to switch (i.e., expenditure-related switching barriers such as the costs of moving one's brokerage depot to another provider; combined effect = +0.65) become more relevant. But psychological and social switching barriers also remain relevant determinants of the dependence dimension. Attractiveness of other brokers, however, has only limited influence on both components of the users' commitment. This might be due to the high costs (including time and money spent) for opening up a new account² at a different online broker.

In summary, it can be concluded that an isolated focus on customer satisfaction will be of limited success, if other determinants of loyalty and/or dependence are neglected. Only a balanced mixture of activities, which simultaneously seek to improve several or all determinants of customer commitment, will have a sustainable positive effect on loyalty and its proposed behavioral consequences.

² For multi-depot-customers (37% of surveyed customers), the effect is expected to be even stronger since they can more easily switch between various providers.

3.2 Behavioral Consequences of Online Customer Commitment

Again, the accepted model was selected based on overall fit to the data as well as the strive to estimate as few parameters as needed. Figure 2 shows the best-fitting partial model (based on a chi-square-comparison) for the behavioral consequences of high customer commitment levels in the online brokerage business.

With exception of NFI, all global fit indices show good fit of the model to the underlying data. In the structural model, 9 of 12 tested causal relationships could be verified with the correct sign and a significance level above 95%. Two more path coefficients are only marginally significant, but show the proposed direction, while - again - the relation from dependence to loyalty is highly significant, but with the opposite sign. Thus, not all of the proposed influences on the 5 behavioral intentions can be supported by the data: For example, while the loyalty state exerts a strong direct effect onto the word-of-mouth-intention, it shows no significant influence onto the willingness to uphold the relationship with the current provider.

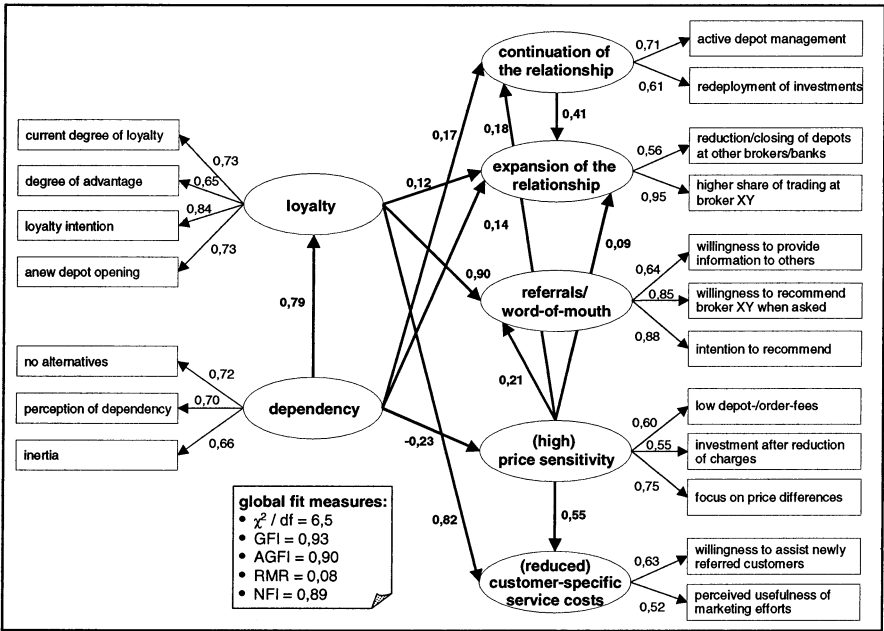


Fig. 2. Best-fitting partial model for behavioral consequences of customer commitment

Table 2 contains the aggregated information about direct and indirect effects as well as their sum. In total, significant effects of the loyalty dimension are only exerted onto the intention for positive reference (+0.90) and their subsequent willingness to support first time users referred by them; leading to a possible reduction of service costs (+0.82). The dependence factor shows similar, but weaker effects on the same two behavioral categories (+0.66 and +0.52), respectively. The strongest direct effect originating from the dependence dimension results in a slightly

reduced price sensitivity of the online brokerage user (-0.23). Interestingly, neither the loyalty nor the dependence dimension exert a significant effect on the intention of an online brokerage user to continue using his current service provider (not significant and +0.13, respectively). An explanation for this seemingly paradox result could be that investment decisions of brokerage users are mainly driven by their expectations about the performance of stocks or bonds; and only to a minor extent by their loyalty towards their preferred online broker. The same reasoning also applies to the online brokerage user's intention of expanding the relationship.

commitment dimension	customer behavioral intention	direct effect	indirect effect(s)	total effect
loyalty	continuation of relationship	--	--	--
	expansion of relationship	0.12	--	0.12
	referrals/word-of-mouth	0.90	--	0.90
	(high) price sensitivity	--	--	--
	(reduced) service costs	0.82	--	0.82
dependence	loyalty	0.79	--	0.79
	continuation of relationship	0.17	- 0.04	0.13
	expansion of relationship	0.14	0.13	0.27
	referrals/word-of-mouth	--	0.66	0.66
	(high) price sensitivity	-0.23	--	-0.23
	(reduced) service costs	--	0.52	0.52

Table 2. Calculation of total effects of the behavioral consequences of high customer commitment in the online brokerage business

4 Summary

Customer commitment seems to be a relevant management goal in online brokerage since the tested models were able to explain commitment and its behavioral consequences. Beyond that, it was empirically verified that the distinction of two separate commitment dimensions is practically relevant to achieve customer behaviors that positively influence a company's profits. Generally, loyalty has stronger effects on desired behaviors of online customers than dependence. Thus, online brokerage firms should focus on pursuing loyalty instead of "forcing" repeat usage based on exit barriers alone. However, as increased dependence positively influences a customer's loyalty, the development of a certain degree of dependence might effectively augment the loyalty strategy to support sustainable commitment levels of online brokerage users. To build up the desired customer "loyalty", online brokerage providers need to focus on developing trust-based, satisfactory and socially-rewarding business relationships with their profitable customers. Concrete marketing activities need to be analyzed based on segment-specific or individual preferences which goes beyond the scope of this paper. Additionally, the online broker should implement user profitability calculations as foundation for a prioritization of customers and efficient allocation of scarce resources.

References

1. ANDERSON EW, FORNELL C, RUST, RT (1997) Customer Satisfaction, Productivity, and Profitability. In: *Marketing Science*, vol 16, no 2, pp 129-145
2. BACKHAUS K, ERICHSON B, PLINKE W, WEIBER R (2000) *Multivariate Analysemethoden* (9th edn). Springer, Berlin etc.
3. BAGOZZI RP, BAUMGARTNER H (1994) The Evaluation of Structural Equation Models and Hypothesis Testing. In: BAGOZZI, RP (Ed). *Principles of Marketing Research*. Blackwell Publishers, Cambridge (MA), pp 386-422
4. BAGOZZI RP, YI Y (1988) On the Evaluation of Structural Equation Models. In: *Journal of the Academy of Marketing Science*; vol 16, no 1, pp 74-94
5. BASS FM, PESSEMIER EA, LEHMANN DR (1972) An Attitude Model for the Study of Brand Preference. In: *Journal of Marketing Research*, vol IX, pp 93-96
6. BENTLER PM, CHOU CP (1987) Practical Issues in Structural Modeling. In: *Sociological Methods & Research*, vol 16, no 1, pp 78-117
7. BOLLEN, KA (1989) *Structural Equations with Latent Variables*. Wiley-Interscience, New York etc.
8. BRUHN M (2001) Nationale Kundenzufriedenheitsindizes. In: HOMBURG C (Ed) *Kundenzufriedenheit* (4th edn). Gabler, Wiesbaden, pp 149-174
9. CORNELSEN J (2000) *Kundenwertanalysen im Beziehungsmarketing*. GIM, Nürnberg
10. DILLER H (1996) Kundenbindung als Marketingziel. In: *Marketing • ZFP*, vol 18, no 2, pp 81-94
11. EGGERT A (2001) Konzeptionelle Grundlagen des elektronischen Kundenbeziehungsmanagements. In: EGGERT A, FASSOTT G (Eds) *eCRM - Electronic Customer Relationship Management*. Schäffer-Poeschel, Stuttgart, pp 87-106
12. FESTINGER L (1957) *A Theory of Cognitive Dissonance*. University Press, Stanford
13. FISCHER M, HERRMANN A, HUBER F (2001) Return on Customer Satisfaction. In: *Zeitschrift für Betriebswirtschaft*, vol 71, no 10, pp 1161-1190
14. GAUL W, HOMBURG C (1987) Different Approaches to Covariance Structure Analysis. In: *Operations Research Proceedings*, vol 8, pp 405-414
15. GIERING A (2000) Der Zusammenhang zwischen Kundenzufriedenheit und Kundenloyalität. Gabler, Wiesbaden
16. HERRMANN A, HUBER F, BRAUNSTEIN C (2000) Ein Erklärungsansatz der Kundenbindung unter Berücksichtigung der wahrgenommenen Handlungskontrolle. In: *Die Betriebswirtschaft*, vol 60, no 3, pp 293-312
17. HOMBURG C, HILDEBRANDT L (1998) Die Kausalanalyse. In: HILDEBRANDT L, HOMBURG C (Eds) *Die Kausalanalyse*. Schäffer-Poeschel, Stuttgart, pp 15-43
18. HOMBURG C, GIERING A (1996) Konzeptualisierung und Operationalisierung komplexer Konstrukte. In: *Marketing • ZFP*, vol 18, no 1, pp 5-24
19. HOMBURG C, BAUMGARTNER H (1995) Beurteilung von Kausalmodellen. In: *Marketing • ZFP*, vol 17, no 3, pp 162-176
20. HOMBURG C, GIERING A, HENTSCHEL F (1999) Zusammenhang zwischen Kundenzufriedenheit und Kundenbindung. In: *Die Betriebswirtschaft*, vol 59, no 2, pp 174-195
21. KLINE RB (1998) *Principles and Practice of Structural Equation Modeling*. The Guilford Press, New York London
22. KOTLER P (1999) *Kotler on Marketing*. The Free Press, New York
23. PETER, SI (1997) Kundenbindung als Marketingziel. Gabler, Wiesbaden
24. PETERSON RA (1994) A Meta-Analysis of Cronbach's Coefficient Alpha. In: *Journal of Consumer Research*, vol 21, September, pp 381-391
25. REICHELDF FF (2001) *Loyalty Rules!* Harvard Business School, Boston
26. REICHELDF FF (1996) *The Loyalty Effect*. Harvard Business School, Boston

Product Bundling as a Marketing Application

Bernd Stauß and Wolfgang Gaul

Institut für Entscheidungstheorie und Unternehmensforschung
Universität Karlsruhe, P.O. Box 6980, 76128 Karlsruhe, Germany
{bernd.stauss, wolfgang.gaul}@wiwi.uni-karlsruhe.de

Abstract. Product bundling describes an interdisciplinary problem of great importance. It can be used to tailor offers to the demand of consumer segments (marketing), it helps to tackle variety reduction management issues (production), it is based on consumer preferences (data analysis), and it needs combinatorial optimization as solution tool (operations research).

In this paper a new profit-maximizing mixed integer product bundling formulation is presented that works well for modest problem instances. Additionally, a heuristic approach is derived that copes with the situation in a Greedy-like manner for larger problem instances by providing a sequence of monotone increasing lower bounds for the objective function of our product bundling methodology.

1 Introduction

The joint offer of two or more different products or services that are sold at a unique price is referred to as bundling. It was first discussed in the early 1960 in a primary legal context with a main focus on monopolists that attempted to expand their monopoly power into other competitive markets by the use of bundling ([13]). By the time two basically different research streams emerged: A more behavioristic orientated group of authors explored perceptual aspects of bundling and associated consequences on consumer preferences ([17], [9], [3]), another direction of research focused essentially on the determination of optimal bundling and pricing strategies. Like the approach of [1], researchers as [10] or [11] provided a graphical framework that allowed for analyzing different bundling strategies where mostly additive structures of product preferences were assumed (e.g., [1], [12], [11]). Motivations for bundling include exploitation of demand complementary, the already mentioned competitive effects, and the consideration of price discrimination. As a major advantage in terms of product variety many authors discussed the possibility of smoothing out consumer preferences as effect inherent in bundling. Some authors like [2] extended their work to situations where competition among firms is prevalent. However, little work has been done up to now concerning the development of usable decision models and appropriate algorithms for generating optimal bundles and prices, respectively. Main contributions in this area were given by the work of [6] and [16].

Since products can be often viewed as bundles of product characteristics ([15]), product bundling received growing attention in new product development due to its ability to tackle the problem of increasing product variety

([4]). As in many cases firms produce a basic version of a certain product which is offered along with several optional features, product variety is mainly generated by mixing and matching those options ([14]). This is fairly common practice, e.g., in the automobile or computer industry.

The remainder of this paper is structured as follows. Section 2 is intended to give an introduction into the fundamentals of a mathematical formulation for generating optimal bundles and prices, respectively. We discuss characteristics as described in the well-known Hanson, Martin paper ([6]) and explain possible problems that may arise when heuristic approaches are formulated. In section 3 a new model is proposed that enables a proper specification of a new heuristic approach and an example is presented to demonstrate its usefulness. Section 4 concludes the paper.

2 Fundamentals of a Mathematical Formulation of Product Bundling

2.1 Assumptions and Optimization Approach

Some preliminary assumptions and notations are needed to establish an underlying framework for product bundling. Starting point is the set of products J whose elements are also called components of the bundles to be created, $\mathcal{B}_l \subseteq J$ is the set of components contained in bundle l , L is the set of all possible bundles, and I is the set of segments of potential buyers (with $i = 1, \dots, |I|$, $j = 1, \dots, |J|$, and $l = 1, \dots, |L| = 2^{|J|} - 1$). Since each product by itself constitutes a bundle, we assume that the first $|J|$ bundles in L correspond to the respective products. The cost c_l of bundle l is the sum of costs of the products contained in the bundle, i.e. $c_l = \sum_{j \in \mathcal{B}_l} c_j$. Bundle l can be offered to segment i at a price p_{il} . Each segment i is characterized by its size N_i and utilities or reservation prices r_{il} for every possible bundle l , i.e. utility is measured in monetary units via reservation prices.¹ Given prices, reservation prices, and the composition of the bundles, every consumer of segment i wants to maximize her/his surplus. Consumer surplus can be described as the difference between reservation price and actual price of the bundle l : $r_{il} - p_{il}$. We assume that consumers cannot get any surplus if they don't purchase anything and that the benefit obtained from multiple components is zero, because to avoid arbitrage we suppose that a resale of components is not possible. On the other side, we assume that there is free disposal of unwanted components.

Furthermore, consumer choice is denoted by binary variables $\theta_{il} \in \{0, 1\}$ with

$$\theta_{il} = \begin{cases} 1, & \text{if segment } i \text{ buys bundle } l \\ 0, & \text{otherwise.} \end{cases}$$

¹ A reservation price can be interpreted as the maximal price a consumer is willing to pay for a bundle.

Additionally, we define indicator variables $y_{jl} \in \{0, 1\}$ where

$$y_{jl} = \begin{cases} 1, & \text{if } j \in \mathcal{B}_l \\ 0, & \text{otherwise} \end{cases}$$

which we will use in our new approach.

Explicit price discrimination is excluded so that all customers face identical prices for the same bundle (i.e. $p_{il} = p_l, i \in I, l \in L$). Despite the assumptions mentioned up to now we allow for combinations of bundles offered without any additional assembly costs. Without doubt the latter isn't consistent with the general assumption within select-one-out-of-many modeling of consumer choice behavior in traditional marketing models and constitutes a main characteristic of bundling. This fact should therefore receive particular attention in the sequel.

Definition 1. We call a price schedule subadditive if the following conditions hold:

$$p_l \leq \sum_{k \in K} p_k \quad l \in L, K \in \{\tilde{K} \mid \cup_{\tilde{k} \in \tilde{K}} \mathcal{B}_{\tilde{k}} = \mathcal{B}_l\}.$$

Under the above assumptions it can be shown that if a profit maximizing firm knows all relevant reservation prices of their potential customers, then there exists a subadditive price schedule which is optimal. As an immediate consequence one may conjecture that each consumer will purchase at most one of the bundles. A major drawback, however, is the fact that all possible bundles have to be considered explicitly.

Due to page restrictions we use the following MHM (Modified Hanson, Martin [6]) formulation:

$$\text{MHM:} \quad \max \sum_{i \in I} \sum_{l \in L} N_i (p_{il} - c_i \theta_{il}) \quad (1)$$

$$\sum_{\tilde{l} \in L} (r_{i\tilde{l}} \theta_{i\tilde{l}} - p_{i\tilde{l}}) \geq \max\{0, r_{il} - p_l\} \quad i \in I, l \in L \quad (2)$$

$$p_l \leq \sum_{k \in K} p_k \quad l \in L, K \in \{\tilde{K} \mid \cup_{\tilde{k} \in \tilde{K}} \mathcal{B}_{\tilde{k}} = \mathcal{B}_l\} \quad (3)$$

$$p_{il} \geq p_l - M(1 - \theta_{il}) \quad i \in I, l \in L \quad (4)$$

$$p_{il} \leq p_l \quad i \in I, l \in L \quad (5)$$

$$\sum_{l \in L} \theta_{il} \leq 1 \quad i \in I \quad (6)$$

$$p_l, p_{il} \geq 0 \quad i \in I, l \in L \quad (7)$$

$$\theta_{il} \in \{0, 1\} \quad i \in I, l \in L \quad (8)$$

where M is a sufficiently large constant. The constraints allow for the following interpretation: The inequalities in (3) enforce a subadditive price schedule (see also definition 1) and consumer choice behavior resembles a first

choice situation as given by constraint (2), i.e. each consumer will buy that bundle which provides the maximal surplus. Although the model allows for segment specific prices p_{il} each segment that actually buys faces identical prices for identical bundles (because if a bundle l is selected ($\theta_{il} = 1$), (4) and (5) enforce $p_{il} = p_l$, otherwise (for $\theta_{il} = 0$) conditions (2) along with the nonnegativity of p_{il} ensure $p_{il} = 0$). Constraints (6), (7), and (8) are self-explaining. As a solution of MHM we get an optimal set of bundles $L_{opt} = \{l \in L \mid \sum_{i \in I} \theta_{il} > 0\}$ that should be offered together with an optimal price schedule. It is worth mentioning that $|L_{opt}| \leq |I|$ holds as each segment chooses at most one of the bundles.

2.2 Price Subadditivity and Further Implications of MHM

A crucial part of MHM is constituted by constraint (3) that describes and guarantees for the special structure of the solution. However, Gaul, Stauß [5] show that it suffices to examine price subadditivity in terms of restrictions for pairs of bundles only instead of those given by (3).

Up to now no assumptions about the functional form of consumer preferences were made. One can tackle this problem by supposing that reservation prices for all bundles are known. But the availability of such information seems to be somewhat unrealistic given the combinatorial variety induced even by a small number of components. Thus, as it is frequently done in the literature, we suppose that reservation prices are additive, i.e. $r_{il} = \sum_{j \in \mathcal{B}_l} r_{ij}$. Furthermore, this assumption enables the use of decompositional methods like conjoint analysis based on linear additive model specifications that provide estimators for reservation prices (see ,e.g., [8], [7] for a discussion of the methodology in the context of product bundling).

An essential drawback of MHM is given by the fact, that for optimization all possible bundles have to be listed explicitly. This is due to the necessity of checking price subadditivity in order to guarantee the appealing structure of the solution.

In the next section a heuristic approach is formulated in which constraints (2) and (3) of MHM are replaced by restrictions that are easier to handle with respect to an incomplete enumeration of bundles.

3 A New Model

3.1 Problem Reformulation

In a current paper of Gaul, Stauß [5] a new formulation of the bundling problem is proposed. The underlying assumptions of the new model are similar to those that were given in section 2 with a slight but essential modification: price subadditivity has not to be postulated explicitly anymore so that a crucial obstacle in developing heuristic approaches is eliminated. On the other

hand the special structure of the solution induced by subadditive prices is still considered.

From now on we will refer to the already introduced indicator notation for bundles y_{jl} instead of the equivalent set representation \mathcal{B}_l and replace constraints (2) and (3) by the following set of conditions:

$$\sum_{l \in L} \left(\theta_{il} \sum_{j \in J} r_{ij} y_{jl} - p_{il} \right) \geq \sum_{j \in J} u_{ij} + \sum_{l \in L} v_{il} + \sum_{j \in J} r_{ij} \quad i \in I \quad (9)$$

$$\sum_{j \in J} u_{ij} y_{jl} + v_{il} \geq -p_l \quad i \in I, l \in L \quad (10)$$

$$u_{ij} \geq -r_{ij} \quad i \in I, j \in J \quad (11)$$

$$-u_{ij} \geq 0 \quad i \in I, j \in J \quad (12)$$

$$v_{il} \geq 0 \quad i \in I, l \in L \quad (13)$$

where u_{ij} and v_{il} are additional variables. As is shown in [5] every optimal solution of MHM is also feasible in the new formulation NBM (New Bundling Model) described by the target function (1) together with constraints (4) - (13).

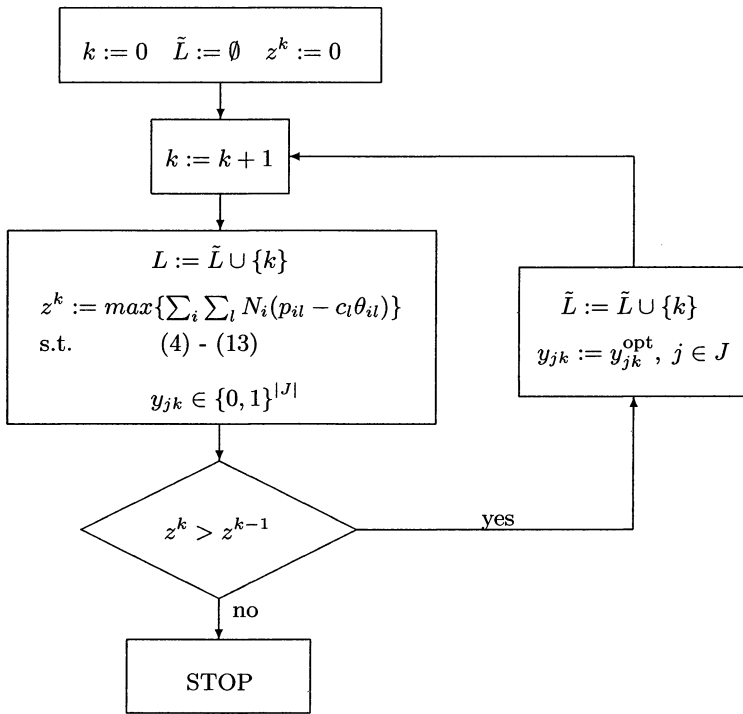
3.2 Heuristic Approach

For NBM the following heuristic approach is suggested. Starting from a given set of bundles $\tilde{L} = \{1, \dots, m\}$ (that could also be empty) bundle augmentation is performed successively. Augmentation step k with $L := \tilde{L} \cup \{k\}$ uses fixed values y_{jl} , $j \in J, l = 1, \dots, k-1$, for the description of bundles determined so far and $y_{jk} \in \{0, 1\}$, $j \in J$, for finding the ‘optimal’ bundle k . Subsequently, the heuristic approach is solved for optimal prices and the bundle candidate k . If the target function (1) can be increased the new bundle (y_{jk}^{opt} , $j \in J$) is added to the set \tilde{L} and the procedure iterates, otherwise the algorithm terminates. This heuristic approach generates a sequence of monotonic increasing lower bounds for (1). It should be noticed that nonlinearities evolve from the introduction of the y_{jk} . However, this is manageable without any difficulties since only one new bundle has to be generated in each step and standard linearization techniques in integer programming can be applied. A flowchart of the heuristic approach is given in Fig. 1.

3.3 Example

Suppose, a firm intends to bundle five products ($j = 1, \dots, 5$) and sell them to six segments ($i = 1, \dots, 6$) of equal size (we use $N_i \equiv 1$, $i \in I$). Segment-specific reservation prices and costs for the products are given in Table 1 from which corresponding values for the bundles can easily be determined.

As the complexity of such problems is mainly dependent on the number

**Fig. 1.** Heuristic Approach for NBM

of segments considered, and the number of products used for bundling and results in considerable computational efforts even for this quite small problem, we used the proposed greedy-like heuristic approach which starts from the empty set of bundles and successively generates at each iteration step

r_{ij}					
(i, j)	$j = 1$	2	3	4	5
$i = 1$	13	15	16	17	23
$i = 2$	14	35	17	10	21
$i = 3$	30	25	9	10	14
$i = 4$	23	10	23	32	19
$i = 5$	13	20	14	40	12
$i = 6$	32	30	27	19	29

c_j					
(i, j)	$j = 1$	2	3	4	5
$\forall i$	29	10	15	10	25

Table 1. Problem parameter

y_{jl}	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	
$j = 1$	0	1	0	0	1	0	
$j = 2$	1	1	1	0	1	1	
$j = 3$	1	0	1	1	1	0	
$j = 4$	1	0	0	1	1	1	
$j = 5$	0	0	0	0	1	0	
	p_l^k						z^k
$k = 1$	62	-	-	-	-	-	108
$k = 2$	62	55	-	-	-	-	124
$k = 3$	65	55	52	-	-	-	133
$k = 4$	69	55	52	55	-	-	141
$k = 5$	74	55	52	55	130	-	153
$k = 6$	74	55	52	55	130	60	154

Table 2. Solution tableau

k a new bundle candidate y_{jk} , $j \in J$, and bundle prices p_l^k ($l = 1, \dots, k$). The solutions at each iteration are tracked by Table 2 that shows the bundle composition, respective prices and how the value of the target function (1) increases.² We further note that y_{j1} , $j \in J$, the first candidate selected in the heuristic approach, has been priced out systematically in the course of the procedure which finally leads to the rejection of this bundle by all customers. Furthermore, segment 1 does not buy at all. It is worth noting that with the proposed bundling strategy a nearly complete skimming of consumer surplus is possible. Only segment $i = 6$ would get a positive surplus from the first bundle but this surplus is too low to keep it from switching to the bundle with the highest surplus.³

Computation time for each iteration of the heuristic approach was within a fraction of a second.⁴

A comparison of the heuristic results and the exact solution obtained via NBM showed that both solutions were identical, i.e., both solution methods provide the same set of bundles that should be introduced. In contrary, if bundling wouldn't be taken into account an optimal pricing schedule for the product would leave the customers much more surplus resulting in a decrease of overall profit for the supplier of the - now - single products by 27%.

² One should mention that in cases in which different bundles would create the same surplus for a segment the heuristic approach selects that bundle with the highest profit for the bundle supplier.

³ In the underlying problem bundle 2 and 5 yield the same surplus of 7 units, thus, the procedure decides for bundle 5 that generates the highest profit.

⁴ The heuristic solution procedure has been implemented using Ilog's Cplex solver. The solver was integrated in an overall Java routine via Ilog's concert interface.

4 Conclusions

In recent years product bundling has become a prevalent marketing practice. However, only a few approaches exist that deal with the determination of optimal bundling and pricing. Promising work was done by Hanson, Martin [6]. But a drawback of their approach is that it requires considerable computational effort and that their mathematical modeling complicates the formulation of heuristic approaches. In this paper, a new approach was presented that allows the application of a simple greedy-like procedure. An example shows how easy it is to get good solutions.

References

1. Adams, W. J., Yellen, Y. L. (1976) Comodity Bundling and the Burden of Monopoly. *Quarterly Journal of Economics* 90, 475–498
2. Chen, Y. (1997) Equilibrium Product Bundling. *Journal of Business* 70, 85–103
3. Estelami, H. (1999) Consumer Savings in Complementary Product Bundles. *Journal of Marketing: Theory and Practice* 7, 107–114
4. Fürderer, R. (1996) Option and Component Bundling Under Demand Risk : Mass Customization Strategies in the Automobile Industry . Wiesbaden : Dt. Univ.-Verl.
5. Gaul, W., Stauß, B. (2003) Designing and Pricing new Bundles. Diskussionspapier No. 255, Institut für Entscheidungstheorie und Unternehmensforschung Universität Karlsruhe
6. Hanson, W. A., Martin, R. K. (1990) Optimal Bundle Pricing. *Management Science* 36, 155–174
7. Jedidi, K., Jagpal S., Manchanda P. (2003) Measuring Heterogeneous Reservation Prices for Product Bundles. *Marketing Science* 22, 107–130
8. Jedidi, K., Zhang, Z. J. (2002) Augmenting Conjoint Analysis to Estimate Consumer Reservation Price. *Management Science* 48, 1350–1368
9. Kaicker, A., Bearden, W. O., Manning, K. C. (1995) Component Versus Bundle Pricing - The Role of Selling Price Deviation from Price Expectations. *Journal of Business Research* 33, 231–239
10. Lewbel, A. (1985) Bundling of Substitutes or Compliments. *International Journal of Industrial Organization* 3, 101–107
11. Salinger, M. A. (1995) A Graphical Analysis of Bundling. *Journal of Business* 68, 85–98
12. Schmalensee, R. (1984) Gaussian Demand and Commodity Bundling. *Journal of Business* 57, 211–230
13. Stigler, G. (1963) United States vs. Loew's Inc: A Note on Block-Booking. *Supreme Court Review* 152, 152–157
14. Swaminathan, J. M. (2001) Enabling Customization Using Standard Operations. *California Management Review* 43, 125–135
15. Telser, L. G. (1979) A Theory of Monopoly of Complementary Goods. *Journal of Business* 52, 211–230
16. Venkatesh, R., Mahajan, V. (1993) A Probabilistic Approach to Pricing a Bundle of Products or Services. *Journal of Marketing Research* 30, 494–508
17. Yadav, M. S. (1994) How Buyers Evaluate Product Bundles: A Model of Anchoring and Adjustment. *Journal of Customer Research* 21, 342–353

Entwicklung und Anwendung einer mehrstufigen Methodik zur Analyse betriebsübergreifender Energieversorgungskonzepte

Wolf Fichtner, Otto Rentz

Institut für Industriebetriebslehre und Industrielle Produktion (IIP),
Universität Karlsruhe (TH), Hertzstr. 16, 76187 Karlsruhe

Zusammenfassung Im vorliegenden Beitrag wird eine Methodik zur Bestimmung der ökonomischen und ökologischen Auswirkungen von betriebsübergreifenden Energieversorgungskonzepten entwickelt und deren Einsatz an einem Praxisbeispiel aufgezeigt. Im Zentrum der Methodik steht ein gemischt-ganzzahliges Optimierungsmodell zur Identifikation der ausgabenminimalen Struktur der betriebsübergreifenden Energieversorgung. Zur Ermittlung der dafür benötigten Parameter wird ein verfahrenstechnisches Prozesssimulationsmodell genutzt und in einer der Optimierung nachgeschalteten Analyse werden die anfallenden Kosten und Erlöse mit Hilfe spieltheoretischer Verfahren unter den Partnern aufgeteilt.

1 Problemstellung und Zielsetzung

Das Schließen von Stoff- und Energiekreisläufen bildet ein Leitmotiv für ein nachhaltigeres Wirtschaften. Wie Sterr [9] aufzeigt, stößt die industrielle Stoffkreislaufwirtschaft auf Industriestandortebene allerdings schnell an ihre Grenzen, weil zumeist der passende Anbieter oder Interessent an einem bestimmten Stoff in der unmittelbaren räumlichen Nachbarschaft fehlt. Während es zur Realisierung stoffkreislaufwirtschaftlicher Potenziale demnach eines über den Industriestandort hinausgehenden Suchraums bedarf, implizieren die Verluste beim Wärmetransport die Forderung nach räumlicher Nähe bei der Kopplung von Energieströmen. Bislang werden Möglichkeiten des betriebsübergreifenden¹ Koppeln von Energieströmen allerdings kaum systematisch analysiert, obwohl aufgrund der mit der Bereitstellung und der Nutzung von Energieströmen verbundenen Emissionen hier noch beachtliche Potenziale zur Reduktion von Umweltbelastungen vermutet werden können. Besondere Relevanz erfährt der Energiebereich auch deshalb weil der durch die Deregulierung des Energiemarktes eingetretene Preiswettbewerb ökologische Maßnahmen verstärkt in den Hintergrund gedrängt hat.

Zielsetzung des vorliegenden Beitrages ist es daher, eine Methodik zur Bestimmung der ökonomischen und ökologischen Auswirkungen betriebsübergrei-

¹ Im Folgenden werden die Begriffe betriebsübergreifend und unternehmensübergreifend vereinfachend synonym verwendet.

fender Vernetzungsoptionen von Energieströmen unter den neuen Rahmenbedingungen des liberalisierten Energiemarktes vorzustellen. Des Weiteren soll der Einsatz dieser Methodik an einem Praxisbeispiel aufgezeigt werden (vgl. auch [3]).

2 Eine Methodik zur strategischen Planung betriebsübergreifender Energiemanagementkonzepte

Die entwickelte mehrstufige Methode zur strategischen Planung betriebsübergreifender Energiemanagementkonzepte besteht aus den Elementen der technischen Analyse, der techno-ökonomischen Optimierung und der Aufteilung möglicher Einsparungen auf die Kooperationspartner. Die technische Analyse hat dabei die Ausarbeitung von konkreten, an die spezifischen Gegebenheiten angepassten Anlagenkonfigurationen als Ziel. Die so erarbeiteten techno-ökonomischen Parameter bilden Eingangsgrößen für die sich anschließende ökonomische Optimierung der Energiebereitstellung. Da ein - so identifiziertes - unternehmensübergreifendes Optimum aus der Sicht eines einzelnen Unternehmens allerdings eine suboptimale Lösung darstellen kann, bedarf es eines dritten Methodenbausteines zur Bestimmung einer stabilen Kosten- bzw. Erlösaufteilung.

2.1 Erster Schritt: Die verfahrenstechnische Prozesssimulation

Um Energiebereitstellungsanlagen auf die spezifischen Rahmenbedingungen anpassen zu können, ist eine spezifische Anlagenauslegung erforderlich. Dabei ist insbesondere die Erfüllung der Energiebilanzen sowie thermodynamischer Gesetzmäßigkeiten zu beachten. Aufgrund der damit verbundenen Komplexität bedarf es zur Bestimmung detaillierter Daten der Simulation der Energieanlagen, wozu die entwickelte Methodik auf das Flow-Sheeting-Simulations-Programm „Aspen Plus“ zurückgreift. Mit Hilfe dieses Simulationsprogrammes werden technische Prozesse in verschiedene unit operations aufgeteilt, bspw. in chemische Reaktionen oder thermische und mechanische Prozesse. Diese unit operations, die die Umwandlung vorgegebener Stoff- bzw. Wärmeströme (Input) in austretende Stoff- bzw. Wärmeströme (Output) beschreiben, werden durch physikalische und chemische Prozesse definiert, die auf thermodynamischen Gesetzmäßigkeiten beruhen. Flow-Sheeting-Simulations-Programme berechnen die Massen- und Energiebilanzen komplex verschalteter verfahrenstechnischer Systeme, basierend auf geeigneten thermodynamischen Modellen. Die Software „Aspen Plus“ verwendet zur Lösung der Massen- und Energiebilanzen der komplex verschachtelten unit operations einen sequentiell modularen Lösungsansatz [1].

Der Vorteil einer solchen verfahrenstechnischen Modellierung liegt darin, dass hierdurch eine adäquate Abbildung der technischen Prozesse erreichbar ist bei gleichzeitig hinreichender Flexibilität hinsichtlich der zu variierenden Betriebs- und Prozessparameter. Dadurch kann die optimale Anpassung neuer Anlagen wie bspw. betriebsübergreifender Kraft-Wärme-Kopplungsanlagen an die lokalen Ge-

gegebenheiten erreicht werden. Die so erhaltenen technischen Auslegungsgrößen stellen die Grundlage für die Bestimmung der ökonomischen Größen dar.

2.2 Zweiter Schritt: Ein Modell zur Identifikation der ausgabenminimalen Energieversorgungsstruktur

Die mit Hilfe der Aktivitätsanalyse [7] mögliche Abbildung von Produktionssystemen mit linearen Input-Output-Beziehungen über einen konstanten Wirkungsgrad und lineare Substitutionsmöglichkeiten bildet die Grundlage für die Technologieabbildung im entwickelten Modell. Mit der Einführung von ökonomischen Größen und Restriktionen wird ein gemischt-ganzzahliges lineares Optimierungsproblem zur systematischen Analyse von (betriebsübergreifenden) Energiemanagementkonzepten formuliert. Das entwickelte Modell PERSEUS-IFC (Program Package for Emission Reduction Strategies in Energy Use and Supply – Inter-Firm Concepts) bildet die energetischen Strukturen der beteiligten Unternehmen detailliert nach. Im Modell werden dazu sowohl existierende Energieanlagen der Unternehmen als auch zukünftige Investitionsalternativen u. a. zur Vernetzung der Energieflüsse anhand techno-ökonomischer Parameter beschrieben. Die Energieanlagen sind durch Energie- und Stoffflüsse sowie über Transport- und Verteilungsanlagen miteinander verknüpft. Ziel des Modells ist es, den ausgabenminimalen Weg zu identifizieren, um die in den Unternehmen benötigten Energieformen bereitzustellen. Die Zielfunktion der Ausgabenminimierung entspricht den realen Gegebenheiten in Industrieunternehmen, da hier die Priorität der eigentlichen Produktionsprozesse zur Konsequenz hat, dass für die Energieversorgung stets das ökonomische Prinzip in seiner Minimumversion gilt.

Des Weiteren wird im Modell eine Vielzahl energiewirtschaftlicher Besonderheiten berücksichtigt, bspw. dass die beim Ausfall eines Kessels wegfallende Leistung kleiner gleich der Summe der restlichen zur Verfügung stehenden Warmreserven und des kontrolliert möglichen Lastabwurfs sein muss. Auf der energetischen Nachfrageseite wird für alle Prozesse der zeitliche Verlauf des Energiebedarfs der Produktionsanlagen mit Hilfe von Lastganglinien approximiert. Dabei werden die Jahre des Betrachtungszeitraumes durch charakteristische Tage repräsentiert, die in mehrere Zeitintervalle unterteilt sind. Das Modell ist in der Programmiersprache GAMS [2] modelliert und benutzt ein auf Microsoft Access basierendes Datenmanagementsystem [5]. Das Planungsproblem lässt sich folgendermaßen formulieren:

Minimiere

$$\sum_{t \in T} \alpha_t \cdot \left(\sum_{f \in F} \left[\sum_{seas \in S} \left[\sum_{p \in P} (Xi_{p,f,t,seas} \cdot Cvar_{p,f,t,seas} + Xo_{p,f,t,seas} \cdot Cvar_{p,f,t,seas}) \right] \right] + \sum_{seas \in S} \sum_{p \in P} PL_{p,t,seas} \cdot Cvp_{p,t,seas} + \sum_{u \in U} [Z_{u,t} \cdot Capn_u \cdot (Cfix_{u,t} + Cinv_{u,t})] \right) \quad (1)$$

unter den Nebenbedingungen

$$\sum_{t'=0}^{t'} rest_{u,t,t'} \cdot newZ_{u,t} = Z_{u,t'} \quad \forall u \in U; \forall t' \in T \quad (2)$$

$$Cpo_{u,t} + Z_{u,t} \cdot Capn_u \geq \sum_{p_u} \left(\frac{PL_{p_u,t,seas}}{h_{seas}} \right) \quad \forall seas \in S; \forall u \in U; \forall t \in T \quad (3)$$

$$Xi_{p,f,t,seas} = \frac{in_{f,p} \cdot PL_{p,t,seas}}{\eta_{p,t}} \quad \forall f \in F; \forall p \in P; \forall seas \in S; \forall t \in T \quad (4)$$

$$PL_{p,t,seas} \cdot out_{f,p} = Xo_{p,f,t,seas} \quad \forall f \in F; \forall p \in P; \forall seas \in S; \forall t \in T \quad (5)$$

$$\sum_{p_{out}} Xo_{p_{out},f,t,seas} \geq D_{f,t,seas} \quad \forall f \in F; \forall seas \in S; \forall t \in T \quad (6)$$

$$\frac{\sum_{p_u} PL_{p_u,t,seas}}{h_{seas}} \leq \sum_{u' \in U \setminus \{u\}} WRES_{u',t,seas} + Labw \quad \forall u \in U; \forall t \in T; \forall seas \in S \quad (7)$$

$$WRES_{u,t,seas} = (Cpo_{u,t} + Z_{u,t} \cdot Capn_u) - \frac{\sum_{p_u} PL_{p_u,t,seas}}{h_{seas}} - HV_{u,t,seas} \quad \forall u \in U; \forall t \in T; \forall seas \in S \quad (8)$$

$$Cpo_{u,t} + Z_{u,t} \cdot Capn_u = HV_{u,t,seas} + HP_{u,t,seas} \quad \forall u \in U; \forall t \in T; \forall seas \in S \quad (9)$$

$$PL_{p,t,seas} \in \mathbb{I}^+ \quad \forall p \in P; \forall seas \in S; \forall t \in T \quad (10)$$

$$Xi_{p,f,t,seas}, Xo_{p,f,t,seas} \in \mathbb{I}^+ \quad \forall f \in F; \forall p \in P; \forall seas \in S; \forall t \in T \quad (11)$$

$$Z_{u,t} \in \mathbb{C}^+ \quad \forall u \in U; \forall t \in T \quad (12)$$

$$WRES_{u,t,seas}, HP_{u,t,seas}, HV_{u,t,seas} \in \mathbb{I}^+ \quad \forall u \in U; \forall t \in T; \forall seas \in S \quad (13)$$

Die zu analysierenden Energiesysteme der beteiligten Unternehmen bestehen aus einer Menge $U = \{1, \dots, m\}$ existierender Anlagen und zukünftiger Zubauoptionen. Jeder dieser Anlagen $u \in U$ ist mindestens ein technischer Prozess $p \in P$ ($P = \{1, \dots, n\}$) zugeordnet, mit dessen Hilfe die Fahrweise der Anlage nachgebildet wird. Diese Prozesse repräsentieren somit den Transformationsprozess des eingehenden zum ausgehenden Energieträger $f \in F$ ($F = \{1, \dots, r\}$). Im Modell wird der gesamte Planungshorizont in einzelne Perioden $t \in T$ ($T = \{0, \dots, w\}$) unterteilt, wobei jede dieser Perioden zur Nachbildung der Energienachfrage mit Hilfe von Lastkurven an charakteristischen Tagen in Zeitintervalle $seas \in S$ ($S = \{0, \dots, x\}$) unter-

gliedert wird. Die Zielfunktion des Modells (1) liegt in der Minimierung der Summe der diskontierten Ausgaben. Dazu sind den Variablen für die Aktivitätsniveaus ($PL_{p,t,seas}$) und für die in einen Energiewandlungsprozess ein- ($Xi_{p,f,t,seas}$) und ausgehenden Energieflüsse ($Xo_{p,f,t,seas}$) Ausgabenkoeffizienten zugeordnet, welche die variablen Ausgaben ($Cvar_{p,f,t,seas}$ und $Cvp_{t,seas}$) charakterisieren. Fixe Ausgabenbestandteile ($Cfix_{u,t}$) und die spezifischen annuierten Investitionen des Kapazitätszuwachses ($Cinv_{u,t}$) sind den Kapazitätsvariablen ($Z_{u,t}$) zugeordnet. Die Gleichungen (2) bestimmen mit Hilfe der Restnutzungsdauer $rest_{u,t,t'}$ die Anzahl an neuen Anlagen $u \in U$, die im Planungszeitraum installiert wurden und in der Periode t noch immer verfügbar sind. Durch die Kapazitätsungleichungen (3) wird sichergestellt, dass in jedem Zeitintervall $seas$ (z. B. Sommerwerktag 8.00 - 10.00 Uhr) jeder Periode t ausreichend Kapazität (bereits vor dem Betrachtungshorizont installierte Kapazität ($Cpo_{u,t}$) plus die Anzahl $Z_{u,t}$ an neuen Blöcken ($Capn_u$)) zur Verfügung steht, um die gewählten Aktivitätsniveaus $PL_{p,t,seas}$ bezogen auf die Intervalllänge h_{seas} erbringen zu können. Mit Hilfe der Nebenbedingungen (4) wird der Input in einen Prozess mit dessen Aktivitätsniveau gekoppelt, wobei der prozessspezifische Wirkungsgrad $\eta_{p,t}$ und der Inputanteil des Prozesses $inf_{p,p}$ berücksichtigt werden. Die Verbindung der Energie- und Stoffflüsse zum Output der einzelnen Technologien wird unter Berücksichtigung des Outputanteils des Prozesses ($out_{f,p}$) über die Gleichungen (5) realisiert. Schließlich garantieren die Nebenbedingungen (6), dass die Energienachfrage $D_{f,t,seas}$ durch entsprechende Flüsse an Energieträgern f in jedem Zeitintervall $seas$ der verschiedenen Perioden t gedeckt wird. Die Ungleichungen (7) stellen für jedes Zeitintervall $seas$ der verschiedenen Perioden t sicher, dass die beim Ausfall eines Kessels wegfallende Leistung kleiner gleich der Summe der restlichen zur Verfügung stehenden Warmreserve $WRES_{u,t,seas}$ und des kontrolliert möglichen Lastabwurfs $LABw$ ist. Die Warmreserve berechnet sich nach den Gleichungen (8) anhand der Differenz aus der vorhandenen Leistung und der im konkreten Zeitpunkt genutzten Leistung. Hierbei muss allerdings berücksichtigt werden, dass die Warmreserve gleich null ist, wenn die Anlage in diesem Zeitpunkt nicht betrieben wird. Hierzu wird in die Gleichungen (8) die Hilfsvariable $HV_{u,t,seas}$ eingeführt, die genau dann den Wert der installierten Kapazität haben muss, wenn die Anlage nicht läuft. In den Gleichungen (9) wird hierfür der Summe der Hilfsvariablen $HV_{u,t,seas}$ und $HP_{u,t,seas}$ der Wert der installierten Kapazität zugewiesen. Auf die des Weiteren benötigten Gleichungen, die unter Rückgriff auf binäre Variable dafür sorgen, dass entweder $HV_{u,t,seas}$ oder $HP_{u,t,seas}$ einen positiven Wert haben, keinesfalls aber beide, sei hier aus Platzgründen verzichtet (siehe hierfür [4]).

2.3 Dritter Schritt: Die nachgeschaltete spieltheoretische Analyse

Falls die identifizierte Optimallösung eine unternehmensübergreifende Lösung darstellt, müssen die Kosten und / oder Erlöse auf die möglichen Kooperationspartner aufgeteilt werden. Insbesondere kann das identifizierte betriebsübergreifende Optimum zu Lösungen führen, die aus der Sicht eines einzelnen Unternehmens eine suboptimale Situation oder gar eine Verschlechterung gegenüber der

Ausgangssituation darstellen. Ein Kooperationsvorhaben ist aber nur dann langfristig stabil, wenn alle beteiligten Unternehmen die Rahmenbedingungen und die Ausgestaltung des Vorhabens akzeptieren. Demzufolge darf keines der beteiligten Unternehmen einen Anreiz besitzen, aus der Kooperation auszuscheren, um sich eventuell im Alleingang oder mit anderen Unternehmen besser zu stellen. Zur Lösung der Allokationsproblematik können die in der BWL existierenden „klassischen“ Verfahren wie Restwert- und Verteilungsrechnung angewendet werden. Um anreizkompatible und langfristig stabile Aufteilungsschlüssel zu entwickeln, sollte alternativ auf Ansätze der kooperativen Spieltheorie zurückgegriffen werden [6]. Hierzu werden im Folgenden vier Konzepte eingesetzt: Den Kern als Vertreter der Mengenkonzepte sowie den Nucleolus, den Shapley-Wert und die Alternate Cost Avoided Method als Vertreter der Wertkonzepte.

3 Anwendung der entwickelten Methodik zur Analyse betriebsübergreifender Energiemanagementkonzepte

Die entwickelte Methodik ist im Rahmen eines BMBF-Projektes zur Entwicklung betriebsübergreifender Energieversorgungskonzepte für drei energieintensive Industrieunternehmen [1,2,3] aus Karlsruhe sowie die dort ansässigen Stadtwerke angewandt worden. Die an dieser Fallstudie beteiligten Unternehmen sind dicht beieinander in der Nähe des Karlsruher Rheinhafens angesiedelt.

Im ersten Schritt der dreistufigen Methodik wurden unter Einsatz des Flow-Sheeting-Simulations-Programms „Aspen Plus“ verschiedene Energiebereitstellungsanlagen ausgelegt. Beispielsweise wurde eine Gas und Dampfturbinen-Anlage (GuD-Anlage-[1,2,3]) zur gemeinsamen Versorgung der drei Unternehmen mit Strom und Dampf konzipiert. Dabei handelt es sich um eine rauchgasseitig dreisträngige Anlage mit drei Gasturbinen sowie drei nachgeschalteten, zusatzbefeueren Abhitzeesseln und einer Gegendruck-Dampfturbine mit Mittel- und Niederdruckdampfauskopplung. Als wesentliche Kenngrößen der Anlage ist eine elektrische Gesamtleistung von 260 MW sowie ein Gesamtwirkungsgrad von ca. 89 % errechnet worden. Die Investitionsschätzung mit Hilfe detaillierter Zuschlagsfaktoren für die Nebenpositionen ergab ca. 198 Mio. €, was einer spezifischen Investition von ca. 760 €/kW_{el} entspricht. Analog erfolgte die techno-ökonomische Charakterisierung sowohl von weiteren betriebsübergreifenden Energiemanagementkonzepten (wie bspw. die Nutzung von Abwärme des Unternehmens 1 in Unternehmen 2) als auch von Investitionsalternativen zur getrennten Energieversorgung bei den beteiligten Unternehmen.

Für die Berechnungen mit Hilfe des PERSEUS-Modells bedarf es des Weiteren gewisser Annahmen hinsichtlich der ökonomischen Rahmenbedingungen. Dazu sind die Energieträgerpreise entsprechend einer Prognose von Prognos [8] angesetzt worden, wobei unterstellt wird, dass die Energieträgerpreise für alle Unternehmen gleich sind, wenn sie die selbe Abnahmestruktur besitzen. Zudem wird für Strom aus Kraft-Wärme-Kopplungsanlagen (KWK-Anlagen), der in ein Stromnetz der öffentlichen Versorgung eingespeist wird, die Zahlung einer Ein-

speisevergütung durch den Netzbetreiber modelliert, die sich aus einem zwischen dem KWK-Anlagenbetreiber und dem Netzbetreiber zu vereinbarenden Strompreis und einem gesetzlich festgelegten Zuschlag zusammensetzt. Schließlich ist die Nachfrageentwicklung in den Unternehmen als konstant angesetzt worden.

Das sich so ergebende Optimierungsproblem besteht aus rund 40.000 kontinuierlichen und ca. 3.000 ganzzahligen Variablen, rund 32.000 (Un-)Gleichungen mit etwa 140.000 Nicht-Null-Elementen. Bei Verwendung eines Rechners mit Intel Pentium III mit 900 MHz Taktfrequenz und einem Arbeitsspeicher von 768 MB können sich Rechenzeiten von bis zu einigen Stunden ergeben, wobei zur Lösung des gemischt-ganzzahligen Optimierungsproblems der kommerziell verfügbare Solver CPLEX 7.0 eingesetzt worden ist.

Die Analysen mit dem PERSEUS-IFC Modell zeigen, dass unter den getroffenen Rahmenbedingungen die dreisträngige GuD-Anlage-[1,2,3] die ausgabenminimale Option zur Befriedigung der Energienachfragen in den beteiligten Unternehmen darstellt. Gegenüber der Referenzentwicklung ohne Zubau können so in den Jahren mit einer KWK-Förderung (d. h. bis 2010) die jährlichen Energieversorgungskosten um ca. 36,1 Mio. €/a reduziert werden; nach dem Auslaufen der KWK-Förderung ist noch eine jährliche Kostenreduzierung von bis ca. 15 Mio. €/a möglich. Hinsichtlich der Anlageneinlastung zeigen die Modellergebnisse, dass die GuD-Anlage-[1,2,3] nahezu im Grundlastbetrieb eingesetzt wird, um Strom und Prozesswärme zu erzeugen. Auftretende Lastspitzen werden mit den in den Unternehmen weiterhin vorhandenen Kesseln und Turbinen abgefahren. Die Bilanzierung der ökologischen Effekte verdeutlicht, dass die Errichtung der GuD-Anlage-[1,2,3] zu einer Minderung sowohl des Primärenergieverbrauchs als auch der betrachteten Emissionen in einer Höhe von ca. 30 % gegenüber der Referenzentwicklung führt. Diese Ergebnisse stützen die Forschungshypothese, dass durch betriebsübergreifende Energiebereitstellungskonzepte ökonomische und ökologische Verbesserungen simultan erzielt werden können.

Um die Realisierung der GuD-Anlage zu ermöglichen, bedarf es allerdings noch der Aufteilung der möglichen Einsparungen auf die beteiligten Unternehmen. In Abb. 1 werden die Ergebnisse der aus der Anwendung des „klassischen“ Restwertverfahrens und der spieltheoretischen Wertkonzepte resultierenden Gewinnaufteilungen dargestellt. Es zeigt sich, dass die aus den spieltheoretischen Ansätzen resultierenden Gewinnaufteilungen eher in der Mitte des Kernlösungsraumes liegen als die Aufteilung nach dem Restwertverfahren. Vergleicht man die spieltheoretischen Ansätze untereinander, so fällt auf, dass die Konzepte des Nucleolus und der ACA-Methode zu einem recht ähnlichen Ergebnis führen. Dagegen werden bei der Ermittlung der Shapley-Werte (Spieler 1: 14,2 Mio. €/a; Spieler 2: 15,3 Mio. €/a; Spieler 3: 6,6 Mio. €/a) verglichen mit den beiden anderen Verfahren die Unternehmen 1 und 2 auf Kosten des Unternehmens 3 besser gestellt. Hier wirkt sich die Tatsache aus, dass Spieler 3 keine Unterkoalitionen bilden kann und somit die gewichtete Summe seiner Grenzbeiträge kleiner ist als die der Spieler 1 und 2. Aufgrund der Eigenschaft der Konvexität des vorliegenden Spiels ist sichergestellt, dass die Shapley-Lösung eine stabile Gewinnverteilung erzeugt. Bei dieser Ausgangslage ist die Shapley-Lösung den anderen untersuchten spieltheoretischen Verfahren vorzuziehen, da sie dann alle wünschenswerten Eigenschaften

(wie Kernzugehörigkeit, Eindeutigkeit, Stabilität, Dummy, strenge Monotonie, Kovarianz, Monotonie im Aggregat, Koalitionsmonotonie, Effizienz, Symmetrie und Additivität [10]) aufweist.

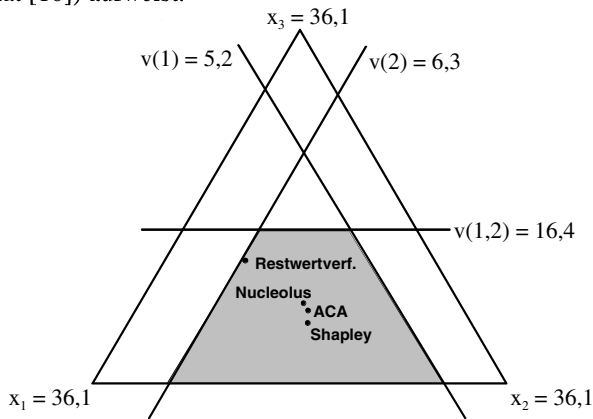


Abb.1: Simplex zur Darstellung möglicher Gewinnaufteilungen (Werte in Mio. €/a)

Literatur

1. Aspen (1994) Aspen Plus™ Getting Started. Aspen Technology Inc, Cambridge, MA
2. Brooke A, Kendrick D, Meeraus A (1998) GAMS – A User's Guide. GAMS Development Corp, Washington DC
3. Fichtner W, Frank M, Rentz O (2003) Strategische Planung betriebsübergreifender Energiemanagementkonzepte. Zeitschrift für Planung 14: 171-196
4. Fichtner W, Wietschel M, Rentz O (2002) Long Term Planning of Inter-Company Energy Supply Concepts. In: OR Spectrum, Special Issue - OR in the Process Industries (Part II): 497-520
5. Göbel M, Frank M, Wietschel M, Rentz O (2001) Berücksichtigung von unsicheren energie- und umweltpolitischen Rahmenbedingungen bei der langfristigen Investitions- und Produktionsprogrammplanung von energietechnischen Anlagen. In: VDI-Gesellschaft Energietechnik (Hrsg.): Fortschrittliche Energiewandlung und -anwendung. VDI Verlag, Düsseldorf
6. Holler MJ, Illing G (2000) Einführung in die Spieltheorie. Springer, Berlin
7. Koopmans TC (1951) Activity Analysis of Production and Allocation. John Wiley & Sons, New York
8. Prognos (Hrsg.) (2000): Energiereport III. Schäffer-Poeschel, Stuttgart
9. Sterr T (2003) Industrielle Stoffkreislaufwirtschaft im regionalen Kontext. Springer, Berlin
10. Wißler W (1997) Unternehmenssteuerung durch Gemeinkostenzuteilung – Eine spieltheoretische Untersuchung. Dissertation. Universität Mannheim

Aspen Plus ist ein Markenzeichen der Aspen Technology, Inc.

A System Dynamics Model of the Epidemiological Transition

Fleßa, S.

Sektion für Gesundheitsökonomik, Universität Heidelberg, Im Neuenheimer Feld 324, 69120 Heidelberg, e-mail: steffen.flessa@urz.uni-heidelberg.de

Zusammenfassung Die Altersstruktur einer Bevölkerung beeinflusst die Empfänglichkeit für Infektionskrankheiten sowie für chronisch-degenerative Erkrankungen. Mit Hilfe eines System Dynamics Modells wird der epidemiologische Übergang von Infektionskrankheiten zu chronisch-degenerativen Erkrankungen simuliert, so dass die Auswirkungen verschiedener Parameter (medizinischer Fortschritt, Alterung etc.) analysiert werden können.

1. Introduction

The age structure of a population is the key for the efficient allocation of health care resources. The population of developing countries is young and mainly suffers from infectious diseases, whereas industrialised countries are facing severe problems of an aging population with many chronic-degenerative diseases. These different diseases call for dissimilar health care systems and for diverse allocation schemes of health care resources. However, disparities are not only characterising the entire world, but even inside one country subpopulations might live in totally different situations. Rural areas of developing countries are commonly populated by many children and women, whereas towns are often the habitat of adult men with completely different diseases. Thus, we are living in many worlds of health - differences which we have to understand in order to make informed health care decisions.

Figure 1 exhibits the model of the so-called demographic transition [1,2,3]. During phase I, the high fertility equals the high mortality, thus there is hardly any population growth. During phase II, the mortality begins to decrease whereas the fertility remains almost unchanged. In some countries fertility even begins to increase. There is a multiplication of the population. The reason for the decreasing mortality can be better hygiene and sanitation or medical progress. The resulting population growth can only continue if additional economic resources for the increased number of people can be provided. This was the case during the European industrial revolution which is frequently seen as a pre-requisite for this phase. During phase III, the mortality continues to decline whereas the fertility is declining as well. Thus, the population density is still increasing. There are various reasons

for the decline in fertility, such as urbanisation, social insurance, changes of cultural values. Finally, in phase IV fertility and mortality have come down to a very low level of about 1-1.5 % and equal each other. Thus, there is no more population increase and in some countries fertility is below replacement reproduction. This is the situation in most western countries. It is assumed that the process of transition (beginning of phase II to end of phase III) takes about 80 years.

The demographic transition is accompanied by a change in disease pattern. Infectious diseases dominate in most countries in the first and second phase of the demographic transition, whereas chronic-degenerative illness is the major cause of morbidity and mortality in developed countries. Although this concept is well known, there have been hardly any attempts to quantify the impact of basic determinants of the demographic transition on the pattern of diseases, although this would be an excellent basis for strategic health care plans. Consequently, a multi-compartment system dynamics model was developed which allows to analyze the epidemiological transition.

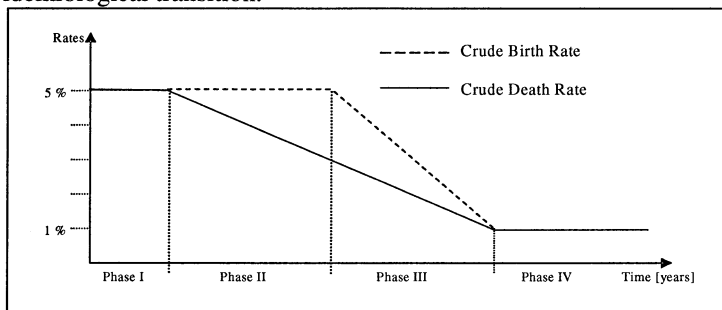


Fig. 1: Model of demographic transition

2. Model

The population of a pattern country is structured in 320 compartments according to age (80 age-sets of each one year) and health status (healthy, infectious diseased, chronic diseased, multi-morbid), i.e.

$$B_{i,j} : \text{population of age } i \text{ in condition } j \quad \text{with } i = 0, \dots, 79 [\text{years}]; j = \begin{cases} 1 & : \text{healthy} \\ 2 & : \text{infectious diseased} \\ 3 & : \text{chronic diseased} \\ 4 & : \text{multi-morbid} \end{cases}$$

It is assumed that 50 % of each compartment are male, 50 % female. The simulation follows fixed time intervals of one day steps. The standard simulation is 120 years, thus: $d := 1..365$ [days/year], $t := 1..120$ [years].

It is assumed that the crude birth rate is merely a given function of time in the demographic transition, i.e. the determinants of the fertility rate are not considered. The crude birth rate is given as:

$$f = \begin{cases} f_begin & \text{for } 0 \leq t < begin_fertil_trans \\ f_begin - \frac{(f_begin - f_end)}{(end_fertil_trans - begin_fertil_trans)} & \text{for } begin_fertil_trans \leq t < end_fertil_trans \\ f_end & \text{for } t \geq end_fertil_trans \end{cases}$$

with

$$\begin{aligned} f_begin & \quad \text{fertility in } t := 0 \\ f_end & \quad \text{fertility in } t := end_fertil_trans \\ begin_fertil_trans & \quad \text{begin of fertility transition} \\ end_fertil_trans & \quad \text{end of fertility transition} \\ t & \quad \text{time} \end{aligned}$$

It is assumed that all children are born healthy, i.e. without infectious and/or chronic diseases. The crude birth rate (f) is a result of the age-specific fertility rates (f_i) with

$$\Delta B_{i,j} = \begin{cases} GEB_t & \text{for } i = 0 \text{ and } j = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$GEB_t = \sum_{i=0}^{79} \sum_{j=1}^4 \frac{1}{2} * f_i * B_{i,j}$$

with

$$f_i = \begin{cases} \frac{f_{29}}{15} * (i - 13) & \text{for } 14 \leq i \leq 28 \\ f_{29} - \frac{f_{29}}{15} * (i - 29) & \text{for } 29 \leq i \leq 43 \\ 0 & \text{otherwise} \end{cases}$$

with

$$f_{29} = 2 * \frac{15 * f * \sum_{i=0}^{79} \sum_{j=1}^4 B_{i,j}}{\sum_{i=14}^{28} i * \sum_{j=1}^4 B_{i,j} - \sum_{i=29}^{43} i * \sum_{j=1}^4 B_{i,j} - 13 * \sum_{i=14}^{28} \sum_{j=1}^4 B_{i,j} + 44 * \sum_{i=29}^{43} \sum_{j=1}^4 B_{i,j}}$$

The simulation of the morbidity distinguishes infectious and chronic diseases. The **immunity** against infectious diseases shall depend on the age.

$$I_inf_{i,j} = \begin{cases} I_inf_max_j * \frac{1}{25} & \text{for } 0 \leq i < 25 \\ I_inf_max_j & \text{for } 25 \leq i < 50 \\ I_inf_max_j * \left(1 - \frac{i-50}{30}\right) & \text{for } 50 \leq i \leq 79 \end{cases} \text{ with}$$

$I_inf_{i,j}$: Immunity against infectious diseases in age i and health status j

$I_inf_max_j$: Maximum immunity against infectious diseases in health status j

$$I_inf_max_j = \begin{cases} 0.9 & \text{for } j = 1 \\ 1.0 & \text{for } j = 2 \\ 0.7 & \text{for } j = 3 \\ 1.0 & \text{for } j = 4 \end{cases}$$

The susceptibility is the contra-event of the immunity, thus:

$$E_inf_{i,j} = \begin{cases} 1 - I_inf_{i,j} & \text{for } j \in \{1,3\} \\ 0 & \text{for } j \in \{2,4\} \end{cases} \quad \text{with}$$

$E_inf_{i,j}$: Susceptibility for infectious diseases in age i and health status j

For chronic diseases the susceptibility is a monotonous function of age.

$$E_C_{i,j} = \begin{cases} 0.2 * E_C_max_j * \frac{i}{35} & \text{for } 0 \leq i < 35 \\ 0.2 * E_C_max_j + \frac{(i-35)}{25} * 0.6 * E_C_max_j & \text{for } 35 \leq i < 60 \\ 0.8 * E_C_max_j + \frac{(i-60)}{30} * 0.2 * E_C_max_j & \text{for } 60 \leq i \leq 79 \end{cases} \quad \text{with}$$

$E_C_{i,j}$: Susceptibility for chronic diseases in age i and health status j

$E_C_max_j$: Maximum susceptibility for chronic diseases in health status j

$$E_C_max_j = \begin{cases} 0.1 & \text{for } j \leq 2 \\ 0.0 & \text{for } j \geq 3 \end{cases}$$

A certain percentage of infectious diseases is prohibited by an increasing standard of general hygiene. The hygiene standard shall be a function of time, i.e.

$$h_t = \begin{cases} 0 & \text{for } t \leq \text{hygiene_begin} \\ h_max * \frac{t - \text{hygiene_begin}}{\text{hygiene_end} - \text{hygiene_begin}} & \text{for } \text{hygiene_begin} < t \leq \text{hygiene_end} \\ h_max & \text{for } t > \text{hygiene_end} \end{cases} \quad \text{with}$$

h_t : hygiene standard in t

hygiene_begin : first year of increasing hygiene standards

hygiene_end : year where hygiene standard reaches its maximum

h_max : maximum hygiene standard

Infectious and chronic diseases are leading to **morbidity** according to:

$$D_{i,2} = B_{i,1} * \frac{\sum_{t=0}^{79} (B_{i,2} + B_{i,4})}{\sum_{i=0}^{79} \sum_{j=1}^4 B_{i,j}} * E_inf_{i,1} * (1 - h_t) * V$$

$$D_{i,3} = B_{i,1} * E_C_{i,j} * V_C$$

$$D_{i,4} = B_{i,3} * \frac{\sum_{i=0}^{79} (B_{i,2} + B_{i,4})}{\sum_{i=0}^{79} \sum_{j=1}^4 B_{i,j}} * E_inf_{i,3} * (1 - h_t) * V + B_{i,2} * E_C_{i,2} * V_C$$

with

$D_{i,2}$: Number of new infectious diseased without chronic diseases in age i

$D_{i,3}$: Number of new chronic diseased without infectious diseases in age i

$D_{i,4}$: Number of new multi - morbid diseased in age i

V : Virulence and contact frequency, calibration variable for infectious diseases

V_C : Calibration variable for chronic diseases

The compartments are adjusted accordingly at the end of each simulation interval. Diseased are **recovering** from their diseases according to:

$$R_inf_{i,j} = \begin{cases} B_{i,j} * \frac{1}{d_inf_j} & \text{for } j \in \{2,4\} \\ 0 & \text{otherwise} \end{cases}$$

$$R_C_{i,j} = \begin{cases} B_{i,j} * \frac{1}{d_chr_j} & \text{for } j \in \{3,4\} \\ 0 & \text{otherwise} \end{cases} \quad \text{with}$$

$R_inf_{i,j}$: Recovery from infectious diseases in age i and heath status j

$R_C_{i,j}$: Recovery from chronic diseases in age i and heath status j

d_inf_j : average length of infectious disease in health status j

d_chr_j : average length of chronic disease in health status j

The compartments are adjusted accordingly.

Three causes of **mortality** are introduced: mortality due to infectious diseases, mortality due to chronic diseases and "rest mortality" in the compartment of age 79. The number of death cases due to the two first reasons is given as:

$$T_inf_{i,j} = B_{i,j} * \frac{m_j}{d_inf_j} * (1 - p_t)$$

$$T_C_{i,j} = B_{i,j} * \frac{m_j}{d_chr_j} * (1 - p_t)$$

$$T_i = T_inf_{i,2} + T_inf_{i,4} + T_C_{i,3} + T_C_{i,4}$$

with

T_i : death cases in age i

$T_inf_{i,j}$: death cases due to infectious diseases in age i and health status j

$T_C_{i,j}$: death cases due to chronic diseases in age i and health status j

m_j : probability to die during the period of disease

p_t : medical progress in t, feasibility of preventing lethal consequences of diseases, with

$$p_i = \begin{cases} 0 & \text{for } t < \text{begin_med} \\ (t - \text{begin_med}) * \frac{p_{\text{max}}}{(\text{end_med} - \text{begin_med})} & \text{for } \text{begin_med} \leq t < \text{end_med} \\ p_{\text{max}} & \text{for } \text{end_med} \leq t \end{cases}$$

begin_med : begin of medical progress

end_med : year where medical progress reaches the maximum

p_max : maximum medical progress

The compartments are adjusted accordingly.

The following **results** are calculated for each year of the simulation: Age distribution, total population, percent of population < 15 years, average age, age-specific mortality rates, life expectancy, percentage of health status 1-4, relation between health status 2:3, prevalence and incidence, death cases due to age, infection, chronic, multi-morbid, under-five-mortality.

3. Results

The standard simulation assumes the parameters as shown in table 1. The basic results are demonstrated in Figures 2 and 3. It is obvious that the health status of the population is a function of time in the demographic transition. About 77.15 % of the population are on average healthy in the first phase of the transition. About 14.20 % of the population are suffering from infectious diseases and 7.60 % from chronic diseases. The rate of multi-morbid diseased (1.05 %) is rather low. The high number of children with an intense susceptibility for infectious diseases leads to a high infant morbidity. The general health improves during the demographic transition until it reaches its maximum at 84.5 % at $t=73$, i.e. the highest percentage of the population is healthy just before mortality and fertility are reaching the fourth phase of the transition. Now a decline starts due to the steadily increasing prevalence of chronic diseases. Finally the population is less healthy (72.32 % in $t=120$) than at the beginning of the epidemiological transition. This deterioration of health is due to the increasing age of the population which leads in the beginning to a higher likelihood of developing chronic diseases. The older the population gets, the more likely infectious diseases will fight their way back to society as very old people are again more susceptible to infectious diseases.

Comparing the percentages of the population suffering from infectious diseases with the share of the population suffering from chronic diseases points at the core of the epidemiological transition: The importance of infectious diseases is getting less whereas the chronic diseases are steadily increasing. In phase I about 14.20 % of the population are sick due to infectious diseases, 7.60 % due to chronic diseases. At the beginning of the demographic transition ($t \geq 20$) infectious diseases are still more significant, but their contribution to the general prevalence of diseases is getting less whereas chronic diseases are getting more. At $t=49$ the two curves are intercepting with 9.02 %. Afterwards the share of chronic diseases is increasing and the share of infectious diseases decreasing.

First year of fertility decrease	60
Fertility in first year	5 %
Fertility at the end of fertility decrease	1 %
First year of hygiene change	20
Last year of hygiene change	100
Maximum share of infectious diseases prevented by hygiene	40 %
First year of medical progress	20
Last year of medical progress	100
Maximum mortality reduction by medical progress	47 %

Tab. 1: Parameter

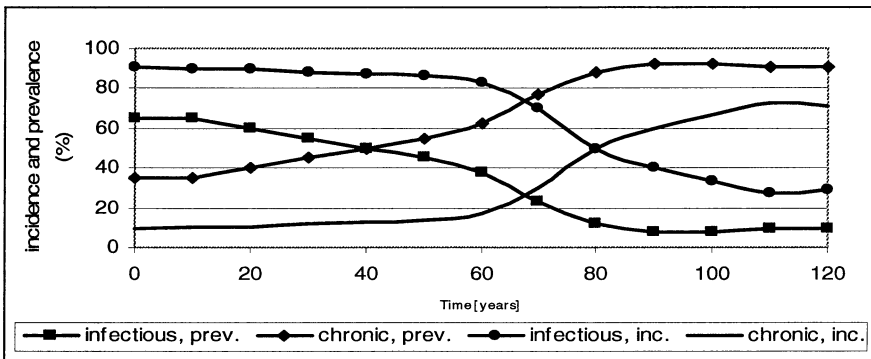


Fig. 2: Prevalence and incidence of infectious and chronic diseases

This result is underlined by Figure 2 which exhibits the prevalence and incidence of chronic and infectious diseases. Here, the prevalence is the relation between the days of sickness due to infectious and chronic diseases. The ratio is 65.15 % to 34.85 % in the beginning of the transition. The shares are equal at $t=49$ and find their maximum distance after the end of the transition ($t=105$) with 5.38 % to 94.62 %. The incidence is the number of new cases not weighted with the length of sickness. As chronic diseases are usually having a longer period of sickness than infectious diseases, the incidence of chronic diseases compared with the incidence of infectious diseases is less at the beginning of the transition (92.45 % infectious to 7.55 % chronic). The pattern of decreasing importance of infectious diseases and increasing importance of chronic diseases is repeated, but the intercept occurs later ($t=87$). This is again due to the fact that patients suffer longer from chronic diseases and therefore a high prevalence is correlated with a low incidence.

The simulations point at an interesting and surprising result: At the end of the simulation period the pattern of the epidemiological transition changes: the share of chronic diseases compared with infectious diseases is decreasing. In other

words: The infections are coming back. This is most likely due to the fact that very old people are more susceptible to infections. Therefore, an aged society is a multi-morbid society with many old people suffering from chronic diseases, others suffering from infections due to their limited immunity, and some are tortured by infectious and chronic diseases at the same time. Whereas the share of people suffering from multi-morbid diseases is merely 0.48 % in $t=80$, it is increasing to 1.09 % in $t=120$. Thus, the aged-society is constituting a severe burden to the economic development of a nation.

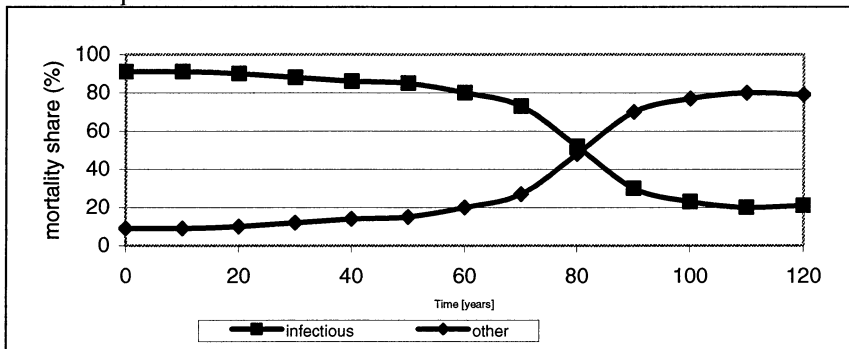


Fig. 3: Mortality: infectious and non-infectious causes

Figure 3 compares the mortality due to infectious and non-infectious (chronic and rest-mortality) diseases. Again we realise that in the beginning 92.07 % of the death cases is due to infections, whereas after 77 years the shares are equal. At $t=110$ the mortality due to infectious diseases is reaching a minimum, afterwards it is going to increase again to 24.08 % in $t=120$. This can be explained in the same way as the increasing morbidity of infectious disease illustrated above.

It becomes obvious that the epidemiological transition is completely reversing the health situation of a country. The different shares of infectious and chronic diseases call for different answers to health care problems and stress the fact that we do not only face different worlds of health status, but also different worlds of health care systems appropriate for different phases of the epidemiological transition. At the same time the data presented above indicates that there is a strong need to analyse the impact of some variables on the epidemiological transition in more details, i.e. the fertility, the sanitation standard and the medical progress.

References

1. Fleßa, S (2002): Gesundheitsreformen in Entwicklungsländern. Lembeck, Frankfurt a. M.
2. Jones, G. W. et al. (1997): The continuing demographic transition. Oxford University Press, Oxford.
3. Miles, D. (1999): Modelling the impact of demographic change upon the economy. The Economic Journal, Vol. 109, No. 452, S. 1-36.

Implementing a Reference Portfolio Strategy in Bond Portfolio Management

Ulrich Derigs¹ and Nils-H. Nickel¹

Department of Information Systems and Operations Research (WINFORS)
University of Cologne
Pohligstr. 1
50969 Cologne, Germany
{derigs, nickel}@winfors.uni-koeln.de

Abstract. In this paper we study a special class of bond portfolio management problems where strategic and operational as well as active and passive aspects are considered. On a strategic level a so-called *reference portfolio* is specified which is valid for a specific product class of bond funds (the class of Eurobond-funds for instance) for a specific period of time. On an operational level this pre-defined mix of assets has to be adapted by the fund managers as precisely as possible for each individual fund on a daily basis respecting all the investment guidelines which have to be fulfilled. Additional lot-size constraints, bid-ask-spreads and commission fees influence the cost of fund restructuring and the associated decision problem leads to a (high-dimensional) discrete non-linear program. We give an outline of a heuristic approach which we have applied in an ongoing decision support system (DSS) development project for a top-european private bank.

1 Introduction

The basic idea of portfolio optimization consists of selecting a subset of assets from an investment universe i.e. an usually very large set of available assets (stocks, bonds etc.) with respect to a given objective representing performance while respecting certain constraints stemming from the budget and from so-called investment guidelines. For the first time this problem was formally modeled by Markowitz (cf. [1]).

While there is a vast number of publications on stock portfolio optimization only few literature exists on bond portfolio optimization models. In this paper we focus on an application of bond portfolio optimization which involves several specific features which are not present in pure stock optimization models as for instance different criteria like the duration etc.

In the application which is the basis of our research we are confronted with a non-homogeneous investment universe, i.e. we distinguish nine different types of bond investments, the called *segments*: government bonds, states/regions, jumbos/mortgage bonds, industrial bonds, Eurobonds of the convergence countries, subordinated debts, asset backed securities and cash. For this reason we cannot apply an evaluation and optimization model which

is designed for only a specific segment but we have to incorporate criteria which are valid for bonds in common.

One main component of any model is the set of investment guidelines. Here we have developed a formal categorization which is the driver for our heuristic approach. In addition to these guidelines, lot-size constraints have to be observed, and bid-ask-spreads and commission fees influence the cost of fund restructuring. Altogether, this framework leads to a (high-dimensional) discrete non-linear program.

2 The Investment Process

The funds of the investment trust company are grouped into several different *product classes* as for instance the class Eurobond-funds etc. In the following we describe the process for one product-class. It involves active as well as passive management on two different levels. On the strategic level a so-called *reference portfolio* is specified which is valid for the specific product class and should guide the fund manager for a specific period of time. On the operational level this pre-defined mix of assets has to be adapted by the fund managers as precisely as possible for the different individual funds on a daily basis respecting all the fund-specific investment guidelines. For the purpose of evaluating the performance of the fund (and the fund managers) with each fund is associated a specific market index, for instance the Lehmann Brother Euro Aggregate, as so-called *benchmark*. Over time sell and buy transactions have to be performed in order to maintain the portfolio feasible with respect to the guidelines as well as to keep the deviation of the fund from the reference portfolio as small as possible. This daily decision problem on transactions is the focus of our study and DSS. Obviously, the model for the operational problem has to incorporate the intention of the strategic decision. Thus we first have to analyze the process on the strategic level.

2.1 Strategic Problem: Construction of the Reference Portfolio

The construction of the reference portfolio is performed in five successive steps. In the first four steps specific target values for different criteria are specified and in the last step an appropriate portfolio is set up, i.e. a set of assets and shares is fixed. In the following we shortly list the criteria for the four steps since these criteria are also relevant for our model for the operational level.

- *Criterion 1: Interest rate risk:* Here a target duration of the reference portfolio and the percentage of investment in each of a set of pre-defined segments of the yield curve is specified.
- *Criterion 2: Credit risk:* Here the portion of investment in non-government bonds (i.e. credits) is specified in total as well as for each segment on the credit yield curve.

- *Criterion 3: Segment*: Here the investment share for the different segments is specified.
- *Criterion 4: Rating*: The financial standing of the portfolio is specified in terms of a pre-defined average rating.

2.2 Operational Level: Approximating the Reference Portfolio

For each specific fund there exist specific investment guidelines which have to be fulfilled. This aspect has not been observed on the strategic level and thus it is usually not feasible to implement the proposal of the reference portfolio. Therefore other shares and other assets of the asset universe have to be selected. Here we can distinguish three classes of investment guidelines: *Legal guidelines* are formulated in national capital investment laws like the *German law on investment trust companies* (KAGG) and they have to be obeyed by every investment fund. For our application mainly §8a KAGG is relevant for the bond portfolio optimization. *Contractual guidelines* reflect the specific investment policy of the fund often negotiated with the investor and are obligatory for this specific fund only. *Internal guidelines* reflect the policy of the firm or the group of fund managers or are set by the financial market, respectively. These constraints may be more "soft" in nature, i.e. they may be violated by the fund manager temporarily. Here, we are mainly confronted with the following constraints:

1. *Lot-sizes* restrict the order value for each asset to a positive multiplier of a pre-defined amount (hard constraints).
2. *Budget Constraints* restrict the net asset value (NAV) of the portfolio. Note that through transactions this value is reduced by the commission fees of the orders (hard constraints).
3. *Cardinality constraints* restrict the number of assets in a portfolio to fall into a specific interval (soft constraints).
4. *Bounds on industrial sectors* give for each industrial sector a maximum allowable deviation from the reference portfolio (hard/soft constraints).
5. *Rating and Liquidity constraints* restrict the average ratings and the shares for investments in the different segments (hard/soft constraints).

Note that the last two classes of internal guidelines are representing aspects of the model guiding the construction of the reference portfolio. In the next section we state an optimization model for generating transaction proposals which ensure the construction or restructuring of portfolios with respect to the investment guidelines which "imitate" the intentions of the reference portfolio. In addition to the strategic criteria which define a measure of distance of a portfolio (*the solution portfolio*) to the reference portfolio we introduce into the objective of our optimization model an excess return measure with respect to the benchmark portfolio as well as transaction costs for restructuring the *actual portfolio*. In figure 1 the relationship of these concepts is summarized.

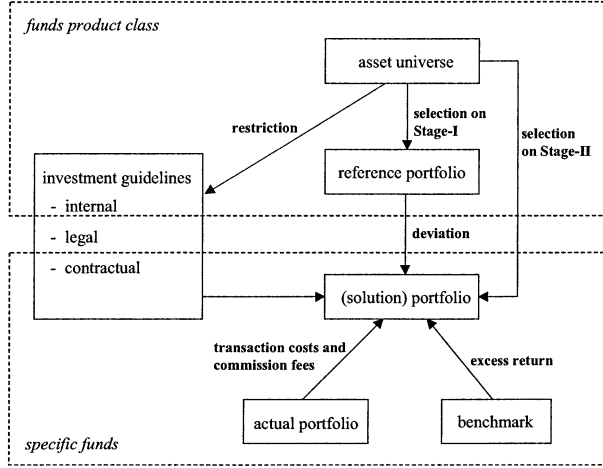


Fig. 1. Relationship and interaction of the different concepts

3 The Optimization Model

Let I the asset universe and $N := |I|$ the cardinality of the asset universe. Then I is partitioned into three sets: I_A the set of bonds, I_F the set of interest rate futures and one (virtual) cash asset i_c . For each asset $i \in I$ the value $y_i \in \mathbb{R}$ gives the *nominal value* in the portfolio and vector $y = (y_1, \dots, y_N)$ represents a (solution) portfolio. Analogously we define the actual portfolio as $\bar{y} = (\bar{y}_1, \dots, \bar{y}_N)$, the reference portfolio as $\hat{y} = (\hat{y}_1, \dots, \hat{y}_N)$ and the benchmark portfolio as $\tilde{y} = (\tilde{y}_1, \dots, \tilde{y}_N)$. Note that for algorithmic purpose i.e. during a local search for instance the use of *shares* for representing portfolios instead of nominal values is more usual and appropriate.

With every asset in the universe we associate the following values:

- $l_i \in \mathbb{Z}_+$ lot-size for asset $i \in I_A$ to be met when buying/selling,
- P_i^G, P_i^B bid resp. ask price of asset $i \in I$ (where the percentage quotations are eliminated by dividing through 100),
- c_i^+, c_i^- commission fees for buying/selling.

For given nominal values y_i ($i \in I \setminus I_C$) the value $MV_i = y_i P_i^G$ gives the so-called *current value or market value*. $NAV(y)$ denotes the *net asset value*, i.e. total market value of portfolio y .

Assume that y is constructed from \bar{y} by a sequence of orders then the cash (market) value can be calculated as

$$MV_{i_c} = \bar{y}_{i_c} \cdot 1 + \underbrace{\sum_{i \in I_A} \max\{0, \bar{y}_i - y_i\} P_i^G}_{\text{sale}} - \underbrace{\sum_{i \in I_A} \max\{0, y_i - \bar{y}_i\} P_i^B}_{\text{purchase}}$$

$$\underbrace{-\left(\sum_{i \in I_A} c_i^- \cdot \max\{\bar{y}_i - y_i\} P_i^G + \sum_{i \in I_A} c_i^+ \cdot \max\{y_i - \bar{y}_i\} P_i^B\right)}_{\text{commission fees}} \quad (1)$$

From an algorithmic point of view the classification of guidelines stated in the last section is irrelevant. In a first formal analysis we have grammaticized all guidelines occurring within the investment company and we have developed a categorization which enables a standard machine-interpretable representation and flexible algorithmic treatment of the vast universe of guidelines. Every guideline can be thought of being composed from a function g on the set of portfolio vectors and a threshold value. The basic concept of our categorization is that of a *bundle* i.e. a specific subset of assets. Here we distinguish the following types of guideline-constraints:

1. *Weighted bundle constraints (WB)* restrict the weighted sum of the market values over all assets in the bundle relative to the NAV of the portfolio as threshold.
2. *Conditional bundle constraints (CB)* are weighted bundle constraints which have to be fulfilled if and only if for a (possibly different) bundle the turnover relative to specific portfolio (usually the current portfolio) exceeds a specific threshold value.
3. *Turnover bundle constraints (TB)* restrict the turnover rate for a specified bundle of assets relative to a second portfolio (usually the current portfolio).
4. *Dynamic bundle constraints (DB)* restrict the sum of the function values over a specified set of non-violated weighted bundle constraints.
5. *Exclusion constraints* exclude the assets of a specified bundle from the asset universe, i.e. the associated nominal values have are fixed to zero.
6. *Cardinality constraints (CC)* limit the number of assets from a specified bundle which are allowed to enter the portfolio.
7. *Fixed-value constraints (FV)* require that the nominal value of a specified asset attains a fixed value, i.e. remains unchanged for instance.

Now let G be the set of all investment guidelines, then we partition G into $G = G^{WB} \cup G^{CB} \cup G^{TB} \cup G^{DB} \cup G^{CC} \cup G^{FV}$. Given a guideline(-function) g and a portfolio y we define a boolean function $Ful(g, y)$ which attains the value 1 if g is fulfilled by y and 0 otherwise and function $Pen(g, y) \in \mathbb{R}_+$ quantifies the violation with respect to the threshold if $Ful(g, y) = 0$ and is set to 0 else.

The objective function of our optimization model is a linear combination of three components which measure the distance of the solution portfolio to the current portfolio, the reference portfolio and the benchmark portfolio:

- $ER(y, \tilde{y})$ is the estimated excess return of y over \tilde{y}
- $DIST(y, \hat{y})$ is the deviation of y to \hat{y} in terms of the criteria of the strategic level

- $TAC(y, \bar{y})$ are the transaction costs for implementing y starting from \bar{y} .

Now we can state the model on an abstract level as follows:

$$\min -\mu_R \cdot ER(y, \tilde{y}) + \mu_D \cdot DIST(y, \tilde{y}) + \mu_T \cdot TAC(y, \bar{y}) \quad (2a)$$

$$s.t. Ful(g, y) = 1 \quad (\text{for all } g \in G) \quad (2b)$$

$$\sum_{i \in I \setminus I_F} MV_i = NAV(\bar{y}) \quad (2c)$$

$$\bar{y}_i - y_i = \alpha_i l_i \quad \text{with } \alpha_i \in \mathbb{Z} \quad (\text{for all } i \in I) \quad (2d)$$

(2c) is the budget constraint and (2d) represents the lot-size constraints. Note that the specification of the μ -parameters in (2a) is part of the organizational implementation and institutionalization of the DSS which is based on this model.

4 The Heuristic Approach

Within the DSS which we have developed for assisting the fund managers we apply a local search based metaheuristic approach for solving the optimization model by adapting and extending the DSS-generator PM-DSS which we have developed and applied for stock management problems (see [2]). As already described in [2], [3] the set of guideline constraints G is partitioned into three classes, i.e. $G = G^{check} \cup G^{pen} \cup G^{move}$ with the following intention: Each guideline $g \in G^{pen}$ is relaxed in a Lagrangean fashion, i.e. the individual violations $Pen(g, y)$ are penalized and added to the objective function as penalty term. Each guideline $g \in G^{check}$ is treated by a check procedure, which evaluates the individual functions $Ful(g, y)$. This information is then used to eventually reject a tentative move during the local search. All guidelines $g \in G^{move}$ are fulfilled within the heuristic search process through an appropriate neighborhood topology. Note that this partitioning is dependent on the actual set of guidelines which are relevant for the specific fund.

In the following we outline the design of the neighborhood topology/move-operator for the local search procedures. We have already argued in the stock management application where the objective function as well as the system of constraints is less complex than the use of "simple moves" which allow a fast evaluation is essential with respect to computational feasibility (see [3]).

For this complex model the neighborhood definition is based on the so-called *1-asset move* operator $y \mapsto y'$. Here we alter the nominal value y_i for a specific asset i by a multiple $\delta \in \mathbb{Z}$ of the corresponding lot-size and we simply compensate this change by a corresponding modification of the cash-asset for which no lot-size constraint has to be considered:

$$y'_i := y_i + \delta \cdot l_i \quad \text{with } \delta \in \mathbb{Z} \text{ for an } i \in I_A \cup I_F \quad (3)$$

Thus this operator has two parameters, the asset i and the discrete *step size* δ . Let y_c denote the current nominal value of the cash asset then the following holds:

$$\begin{aligned}
 NAV(y') &= \sum_{j \in I} y'_j P_j^G = \left(\sum_{j \in I_A \setminus \{i\}} y_j P_j^G + y'_i P_i^G \right) + y'_c \\
 &= \sum_{j \in I_A} y_j P_j^G + \delta l_i P_i^G + y'_c \\
 &= NAV(y) + \delta l_i P_i^G + f_i(\delta)
 \end{aligned} \tag{4}$$

The function $f_i(\delta)$ describes the change in the nominal value of the cash asset depending on the choice of δ . Here, we have to distinguish two cases:

- (1) $y_i - \bar{y}_i \geq 0$, i.e. a purchase of asset i (relative to \bar{y})
- (2) $y_i - \bar{y}_i < 0$, i.e. a sale of asset i (relative to \bar{y})

For both cases $f_i(\delta)$ is a continuous monotonically decreasing function composed from two linear functions

$$f_i(\delta) = \begin{cases} c_1 \delta + (b - c_1 a) & ; \quad \delta \leq a \\ c_2 \delta + (b - c_2 a) & ; \quad \delta \geq a \end{cases} \tag{5a}$$

$$= \max\{c_1 \delta - c_1 a, 0\} + \min\{c_2 \delta - c_2 a, 0\} + b \tag{5b}$$

with $c_1, c_2 < 0$.

We have developed formulas for all the above stated constraints/guideline classes which given an asset i determine the intervall/set of feasible step sizes δ . During the search we first select the parameter i for a move and we successively check all guidelines in order to determine the set of feasible step-sizes, i.e. the set of δ -values such that each guideline $g \in G^{move}$ is fulfilled. Then we determine a feasible step-size according to a specific criterion based on the the change in the objective function.

We have embedded this search strategy into a *Best First*, a *Simulated Annealing* and a *Tabu Search* meta-heuristic (cf. [4]). Note that the use of the simple 1-asset move operator allows a highly flexible application of the search strategies, obviously the class of two-asset-moves which we have introduced for the tracking error minimization model in stock management can easily be simulated. For solving the optimization model we apply a hybrid multi-stage approach which we outline in the sequel:

- *Phase 0: (Preprocessing):*

In a preprocessing phase we eliminate those assets from the asset universe for which exclusion constraints are valid and we analyze the set of guidelines and eliminate constraints which are dominated by other constraints. Then in an optional next preprocessing step we formulate a relaxation of our optimization model incorporating the lot-size and budget constraints as well as all weighted bundle, turnover bundle and fixed

value constraints. The resulting mixed integer program is then solved by a standard MIP-solver. The intention of this optional step is to generate a good starting solution for the subsequent search, to identify more redundant constraints and eventually to proof infeasibility.

- *Phase 1* (Generating feasibility):

Here local search is started either from the solution of the MIP-problem or from the current portfolio \bar{y} . In the latter case, we first have to construct a solution which fulfills the fixed value constraints by performing appropriate moves. Then we apply a local search heuristic with the partitioning $G^{check} := \emptyset$, $G^{move} := \{g \in G^{WB} \cup G^{TB} \cup G^{CB} \cup G^{DB} \mid Ful(g, y) = 1\}$, $G^{pen} := G \setminus G^{move}$ and the auxiliary objective function $F(y) := \sum_{g \in G^{pen}} Pen(g, y)$ until a feasible solution is obtained.

- *Phase 2* (Improving while maintaining feasibility):

After phase 1 returns a feasible solution we apply a local search heuristic with the following partitioning $G^{check} := \emptyset$, $G^{move} := G$, $G^{pen} := \emptyset$ and the original objective function of the optimization model.

Note that in phase 1 and 2 we may apply different local search procedures. Such a flexibility is appropriate since the constraints as well as the neighborhoods are different in both phases.

5 Final Remarks

Within a still ongoing research project we have developed in cooperation with a leading German investment trust company a personalized decision support system which is based on this approach and which serves to assist the fund managers within bond fund management. The implementation was performed by customizing PM-DSS, a DSS-generator for portfolio optimization, which has been developed at our institute and which has already been applied successfully to problem instances in passive stock fund management (cf. [2] or [3]). In this paper we have described the conceptual aspects of the application to bond portfolio management only, a detailed description of the implementation as well as results will be given in subsequent papers.

References

1. Markowitz, H. (1952) Portfolio Selection. *Journal of Finance*, March, 77–91.
2. Derigs, U. and Nickel, N.-H. (2003) Meta-heuristic Based Decision Support for Portfolio Optimization with a Case Study on Tracking Error Minimization in Passive Portfolio Management. *OR Spectrum* 25 (3), 345–378
3. Derigs, U. and Nickel, N.-H. (2003) On a Local-Search Heuristic for a Class of Tracking Error Minimization Problems in Portfolio Management. to appear in: *Annals of Operations Research*, Special Issue on Metaheuristics (Theory, Applications and Software)
4. Aarts, E. and Lenstra, J.K. (1997) *Local Search in Combinatorial Optimization*. John Wiley & Sons

Evolutionary Algorithms and the Cardinality Constrained Portfolio Optimization Problem

Felix Streichert, Holger Ulmer, and Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72076 Tübingen, Germany,
`streiche@informatik.uni-tuebingen.de`

Abstract. While the unconstrained portfolio optimization problem can be solved efficiently by standard algorithms, this is not the case for the portfolio optimization problem with additional real world constraints like cardinality constraints, buy-in thresholds, roundlots etc. In this paper we investigate two extensions to Evolutionary Algorithms (EA) applied to the portfolio optimization problem. First, we introduce a problem specific EA representation and then we add a local search for feasible solutions to improve the performance of the EA. All algorithms are compared on the constrained and unconstrained portfolio optimization problem.

1 Introduction

Evolutionary Algorithms (EA) have been successfully applied to many optimization problems in science and technology. Some researchers also used EA on financial engineering problems like the portfolio optimization problem [4,1,8]. In this paper we compare the impact of several EA solution representations and the application of local search on the performance of EA on the portfolio optimization problem.

2 The Portfolio Optimization Problem

Using the standard Markowitz mean-variance approach [6], the unconstrained portfolio optimization problem is given as

$$\text{minimizing the variance of the portfolio : } \sum_{i=1}^N \sum_{j=1}^N w_i \cdot w_j \cdot \sigma_{ij}, \quad (1a)$$

$$\text{maximizing the return of the portfolio : } \sum_{i=1}^N w_i \cdot \mu_i, \quad (1b)$$

subject to

$$\sum_{i=1}^N w_i = 1, \quad (2a)$$

$$0 \leq w_i \leq 1 \quad ; \quad i = 1, \dots, N \quad (2b)$$

where N is the number of assets available, μ_i the expected return of asset i , σ_{ij} the covariance between asset i and j , and finally w_i are the decision variables giving the composition of the portfolio.

The optimization problem as given in Equ. 1 is a multi-objective optimization problem with two competing objectives. First, to minimize the variance (risk) of the portfolio and at the same time to maximize the return of the portfolio. Equ. 2 gives the minimum constraints for a feasible portfolio.

While this portfolio optimization problem is a quadratic optimization problem for which computationally effective algorithms exist, this is not the case if real world constraints are added:

Cardinality Constraints restrict the number of assets in the portfolio.

$$\sum_{i=1}^N \text{sign}(w_i) = k \quad (3)$$

Buy-in Thresholds give the acquisition prices for each asset.

$$l_i \leq w_i; \quad i = 1, \dots, N \quad (4)$$

Roundlots give the smallest volumes f_i that can be purchased.

$$w_i = y_i \cdot f_i; \quad i = 1, \dots, N \text{ and } y_i \in \mathbf{Z} \quad (5)$$

Other real world constraints can include sector/industry constraints, immunization/duration matching and taxation constraints, but these will not be addressed in this paper.

3 The Optimization Algorithm

To solve the multi-objective optimization problem we use a Multi-Objective Evolutionary Algorithm (MOEA) having two different EA representation types with an additional problem specific extension for each representation. To further improve the results we apply a local search for feasible solutions. There are two approaches how to incorporate local search into EA (Memetic Algorithms) either using Lamarckism or relying on the Baldwin effect [10].

3.1 Evolutionary Algorithms

EAs are population based stochastic optimization heuristics inspired by Darwin's Evolution Theory. An EA searches through a solution space in parallel by evaluating a set (population) of possible solutions (individuals). An individual gives a solution by representing the decision variables w_i .

An EA starts with a random initial population P_0 . Then the 'fitness' of each individual is determined by evaluating the objective function, Equ. 1. After the best individuals P'_t are selected, new individuals for the next generation P_{t+1} are created from P'_t . New individuals are generated by altering the individuals of P'_t through random mutation and by mixing the decision variables of multiple parents through crossover. Then the generational cycle repeats

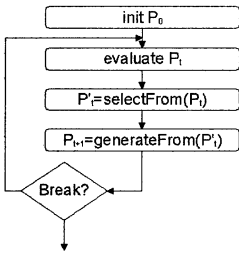


Fig. 1. EA scheme

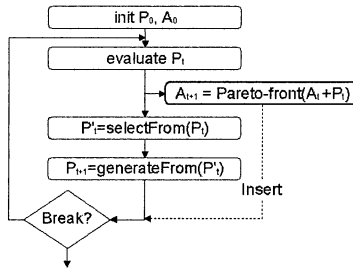


Fig. 2. Multi-Objective EA

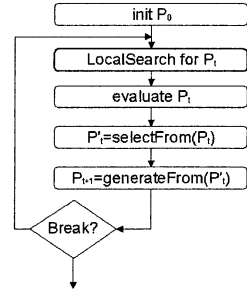


Fig. 3. Memetic Alg.

until a breaking criteria is fulfilled, see Fig. 1 for a basic scheme.

There are several alternative EA implementations but only two will be considered in this paper: Genetic Algorithms (GA) [5] are based on a binary representation for real value decision variables and rely on big populations and the crossover operation. Evolutionary Strategies (ES) [9] on the other hand use a direct real value representation and apply sophisticated mutation operators. An extensive overview on EAs can be found in [2].

3.2 Multi-Objective Evolutionary Algorithms

Due to the population based search strategy and the simple selection strategy EAs are easy to extend to multi-objective optimization problems. First, by using selection based on multiple objective values like the Pareto-dominance criteria. Secondly by adding an archive population A_t used to maintain the currently known Pareto-front. Zitzler gives a guide to MOEAs in his Ph.D. Thesis [12].

During multi-objective optimization two goals are to be reached. On the one hand the solutions should be as close to the global Pareto-optimal front as possible and on the other hand the solutions should also cover the whole Pareto-front. The first goal is often achieved through elitism by replacing random individuals in P_t with individuals on the Pareto-front A_t , see Fig. 2. The second goal can be achieved by punishing individuals that are too close together (Fitness Sharing).

3.3 Memetic Algorithms

Memetic Algorithms (MA) [7] extend EA by adding an arbitrary (possibly problem specific) local search heuristic before evaluating the population P_t , see Fig. 3. There are two alternatives of how to integrate the local search [10], first by updating only the enhanced objective values (fitness) for each individual (Baldwin Effect) or by also updating the decision variables, so that they can be inherited to the next generation (Lamarckism). An example for MA on the portfolio optimization problem is given in [8].

4 Experimental Settings

In our experiments we apply a generational GA population strategy with a population size of 500 individuals. We use tournament selection with a tournament group size of 8 together with objective space based fitness sharing with a sharing distance of $\sigma_{share} = 0.01$. The selection mechanism prefers individuals that are better than other individuals in at least one objective value, i.e. are not dominated by another individual. To maintain the currently known Pareto-front we use an archive of 250 individuals and use A_t as elite to achieve a faster speed of convergence. Details of this MOEA strategy can be found in [11].

We use two different standard EA representations on the portfolio optimization problem: First, a GA binary representation with a 32 bit standard binary encoding (genotype) for each w_i (phenotype). A 3-point-crossover and one-point-mutation is used on the GA genotype, with a crossover probability of ($P_c = 1.0$) and mutation probability of ($P_m = 0.01$). Secondly we use an ES with a real-valued solution representation. In this case the phenotype equals the genotype. We apply a discrete 3-Point-Crossover ($P_c = 0.5$) and local mutation with one strategy parameter for each decision variable ($P_m = 1.0$) on the ES genotype. These parameters were selected from preliminary experiments.

The idea to use a problem specific representation for the portfolio optimization problem is based on the fact that portfolios on the Pareto-front are rarely composed of all available assets, but only a limited selection of assets. The actual composition of the assets in the portfolio resembles a one-dimensional binary knapsack problem. To allow easy removal and adding of assets to the portfolio we added an auxiliary binary bit-mask b_i together with the decision variables w_i to represent a solution. Each bit b_i determines whether the associated asset will be element of the portfolio or not, $w'_i = b_i \cdot w_i$. Both EAs were enhanced with this additional 'knapsack' representation. The extended EAs will be referred to as Knapsack-GA (KGA) and Knapsack-ES (KES).

The second extension is made to improve the number of feasible solutions generated by the EA. Instead of punishing or rejecting infeasible solutions, we apply a 'local search heuristic' to convert an infeasible solution into a feasible one. For example to hold Equ. 2 we limit the range of the EA solution representation and use a standardization step $w'_i = w_i / \sum_{j=1}^N w_j$. For cardinality constraints we set all but the k biggest decision variables w_i to zero before standardization. Similar mechanisms were applied for buy-in thresholds and roundlot constraints.

5 Results

The comparison of the different EA implementations was performed on benchmark data sets given by Beasley [3]. The numerical results presented here are

performed on the *Hang Seng* data set with 31 assets.

To compare the performance we measure the percentage difference (Δ_{area}) between the area below the Pareto-front generated by the EA and the area below the unconstrained Pareto-front given as reference solution, see Fig. 8. For each experiment 50 independent runs each with 100.000 fitness evaluations were made. For these we calculate the mean, standard deviation, maximum and minimum values and the 90 % confidence interval of the Δ_{area} , which is to be minimized.

5.1 Adding the Knapsack representation to the EA individuals

When comparing the GA and ES against the KGA and KES without additional constraints (no l_i and f_i) the extended versions clearly outperform the standard EA approaches, see Fig. 4. Especially the KES shows very good convergence behavior. Only in case of $k = 2$ the GA and ES are able to catch up with the extended EA representations. This shows that the assumption that the portfolio optimization problem resembles the binary knapsack problem holds true even without cardinality constraints and that the extended representation is able to search more efficiently than the standard EAs.

With additional buy-in thresholds and roundlots ($l_i = 0.1$ and $f_i = 0.02$) all algorithms performed much worse, see Fig. 5. Although single runs of the extended EAs find reasonable good solutions, the results are rather unreliable.

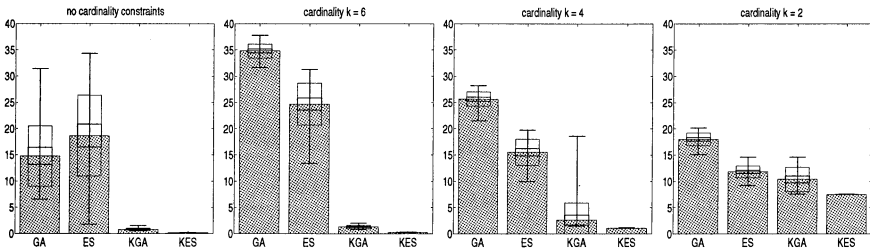


Fig. 4. Δ_{area} for Hang Seng without l_i and f_i

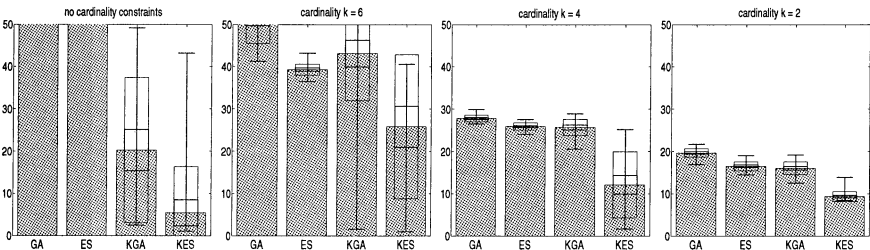


Fig. 5. Δ_{area} for Hang Seng with $l_i = 0.1$ and $f_i = 0.02$

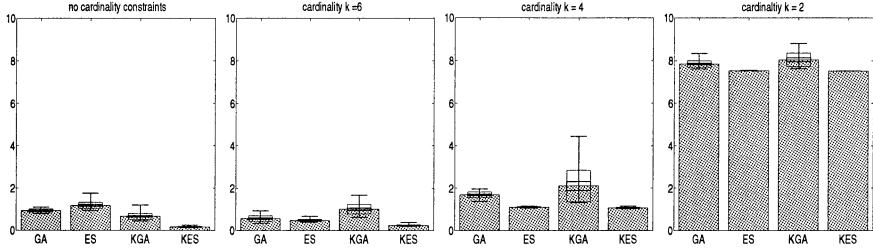


Fig. 6. Δ_{area} for Hang Seng with Lamarckism and without l_i and f_i

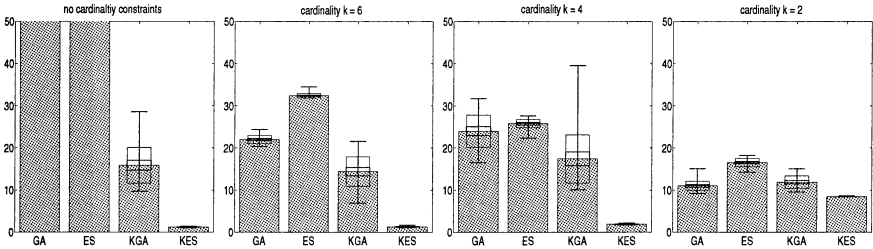


Fig. 7. Δ_{area} for Hang Seng with Lamarckism and $l_i = 0.1$ and $f_i = 0.02$

5.2 Applying Lamarckism

The main advantage of the extended EA seems to be the ability to easily change the content of the portfolio. The same effect could be achieved if local search with Lamarckism is used. With Lamarckism assets removed by local search would cause the decision variables to be set to zero. Therefore the resulting vector of decision variables w_i would be sparse. On such a sparse vector single mutations and crossover could cause major changes of the content of the portfolio. With this, standard EAs could move as easily through the space of assets combinations as the extended EA.

The experiments with Lamarckism supports this view, see Fig. 6. The standard EAs become competitive to the extended EAs and especially the ES behaves nearly as good as the KES except for the unconstrained portfolio optimization. In the latter case, Lamarckism can not have the same effect since no assets are removed through local search.

In case of additional constraints all algorithms perform much better and the results become more reliable, see Fig. 7. Even the extended EAs benefit from the Lamarckism because the extended representation is as improper for the discretization of the search space through the roundlot constraints as the standard EAs are for searching for portfolios with limited cardinality. Remarkably, the KES with Lamarckism seems to perform as well as in the unconstrained case. Most likely the discretization of the Pareto-front through the limited archive size levels the effect of the roundlot constraint.

6 Conclusions and Future Work

We have shown that the extended EA with the additional knapsack representation is able to solve the portfolio optimization problem more efficiently than the standard EA approaches, due to the improved search capabilities regarding the possible combinations of assets in a portfolio. The KES showed to be superior to the KGA due to the more appropriate ES real-value representation of the decision variables w_i . We were also able to produce the same effect by using the Memetic feasibility search in combination with Lamarckism. In this case the standard EAs were able to draw level with the extended EAs and as before the ES with Lamarckism produced better results than the GA with Lamarckism and even better than the KGA.

With additional constraints all algorithms performed not as well except a few good outliers when using the extended EA. Again, Lamarckism is able to improve the algorithms. With Lamarckism the KES reliably produces results only slightly worse than in the unconstrained case. Preliminary experiments hint that a discrete representation for KGA performs much better in case of roundlot constraints and produces equally good results.

Our future research will concentrate on improving the Multi-Objective EA and comparing alternative Multi-Objective EAs on the portfolio selection problem. Another area of improvement could be the local search. There are numerous alternatives to the simple search for feasible solutions, but they have to be carefully evaluated regarding the ability to handle additional constraints.

Acknowledgements

This research has been funded by the German Federal Ministry of Research and Education (BMBF) under contract No. 03C0309E.

References

1. S. Arnone, A. Loraschi and A. Tettamanzi (1993) A Genetic Approach to Portfolio Selection, in *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems* (3)6:597-604
2. T. Baeck, D.B. Fogel, Z. Michalewicz editors (1997) *Handbook of Evolutionary Computation*, Oxford University Press, New York, and Institute of Physics Publishing, Bristol
3. J. B. Beasley (1996) Obtaining test problems via the Internet, *Journal of Global Optimization*, 8:429-433
4. T.-J. Chang, N. Meade, J.B. Beasley and Y.M. Sharaiha (2000) Heuristics for cardinality constrained portfolio optimization, *Computers and Operations Research*, 27:1271-1302
5. J. Holland (1972) *Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Systems*, The University Press of Michigan Press, Ann Arbor

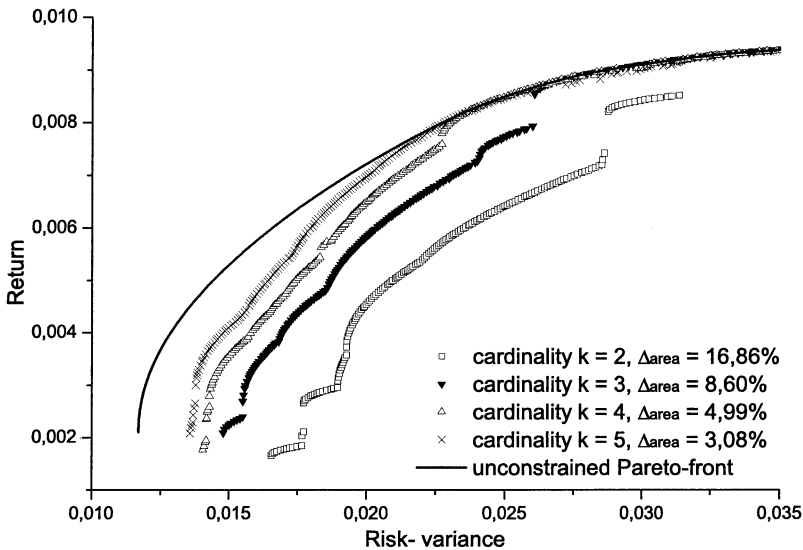


Fig. 8. Solutions generated by EA on the DAX data set

6. H. M. Markowitz (1952) Portfolio Selection, *Journal of Finance*, 1(7):77–91
7. P. Moscato (1989) On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826
8. F. Schlottmann, and D. Seese (2001) A Hybrid Genetic-Quantitative Method for Risk-Return Optimization of Credit Portfolios, *Proceedings of the Conference of Quantitative Methods in Finance*, University of Technology, Sydney, Australia, p. 55
9. H.-P. Schwefel (1995) *Evolution and Optimum Seeking*, John Wiley & Sons, Inc., New York
10. D.L. Whitley and V.S. Gordon and K.E. Mathias (1994) Lamarckian Evolution, The Baldwin Effect and Function Optimization, in *Parallel Problem Solving from Nature (PSN III)*, Y. Davidor and H.-P. Schwefel and R. Männer eds., Springer, pp. 6–15
11. N. Srinivas, Kalyanmoy Deb (1994) Multiobjective Optimization Using Non-dominated Sorting in Genetic Algorithms, *Evolutionary Computation* (2)3:221–248
12. E. Zitzler (1999) *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Swiss Federal Institute of Technology (ETH) Zurich. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Germany, ISBN 3-8265-6831-1

Calculating Concentration-Sensitive Capital Charges with Conditional Value-at-Risk

Dirk Tasche¹ and Ursula Theiler²

¹ Deutsche Bundesbank, Postfach 10 06 02, 60006 Frankfurt am Main, Germany
E-mail: tasche@ma.tum.de

The opinions expressed in this note are those of the authors and do not necessarily reflect views shared by the Deutsche Bundesbank or its staff.

² Risk Training, Carl-Zeiss-Str. 11, 83052 Bruckmühl, Germany
E-mail: theiler@risk-training.org

Abstract. By mid 2004, the Basel Committee on Banking Supervision (BCBS) is expected to launch its final recommendations on minimum capital requirements in the banking industry. Although there is the intention to arrive at capital charges which concur with economic intuition, the risk weight formulas proposed by the committee will lack an adequate treatment of concentration risks in credit portfolios. The question arises whether this problem can be solved without recourse to fully-fledged portfolio models. Since recent practical experience shows that the risk measure Conditional Value-at-Risk (CVaR) is particularly well suited for detecting concentrations, we develop the semi-asymptotic approach by Emmer and Tasche in the CVaR context and compare it with the capital charges recently suggested by the Basel Committee. Both approaches are based on the same Vasicek one-factor model.

1 Introduction

From an economic point of view, the risks that arise in a portfolio need to be covered by a corresponding amount of capital that is applied as a cushion to absorb potential losses. The question of calculating the risk contributions of single assets in a portfolio corresponds to the problem of calculating capital charges to cover occurring loss risks. This allocation issue can be considered as well from a regulatory as well as from an internal perspective, leading to capital charges per asset by the regulatory and the economic capital, respectively.

Considering the different regulations for credit risk, we observe that in the current regulatory regime (Basel I Accord) almost no risk adjustment can be identified. By mid 2004, the BCBS is expected to launch its final recommendations on new minimum capital requirements in the banking industry (Basel II Accord). Although there is the intention to arrive at improved more risk sensitive capital charges of the credit risk bearing assets, the risk weight formulas proposed by the committee will lack an adequate treatment of concentration risks in credit portfolios as we will show below. The main problem

is that the Basle II model assumes that all credits are of equal, infinitesimal small exposure, i.e. that the credit portfolio is infinitely fine grained. However, in real world portfolios, this basic assumption does not hold.

From an internal perspective, banks are putting high efforts into the development of internal credit risk models that allow the risk measurement of the portfolio credit risk. Comprehensive research work has been done to develop methods of how to calculate risk contributions in an appropriate way (see e.g. [4], [7], [9], [11], [12]). However, from a practical point of view, the allocation problem, i.e. the question of how the single assets contribute to the overall portfolio risk, cannot yet be considered as solved. Risks are broken down in different ways, taking into account correlation effects and concentration risk to different extents (see e.g. [3]). Banks that are using internal credit risk models in most cases need enormous calculation efforts to estimate the overall portfolio risk and the risk contributions of single assets and sub-portfolios.

In this paper, we give a survey on an approach to calculate risk contributions of single assets in an analytical way that avoids calculation intensive simulation efforts. This *semi-asymptotic* approach slightly extends the Basel II approach and takes into account concentration effects. Thus, it can be viewed as a bridging to relate regulatory and internal risk measurement. Additionally, it is based on the new risk measure of Conditional Value at Risk that has been proven to be appropriate for bank wide loss risk measurement (see for instance [1], [10], [12]).

The paper is organized as follows. In Section 2 we briefly introduce the capital charges as they are suggested by the Basel Committee. Section 3 presents the semi-asymptotic approach to capital charges in case of CVaR as risk measure. In Section 4 we illustrate the both approaches with a numerical example.

2 The Basel II Model

We give a short presentation of the reasoning that lead to the current suggestions by the Basel Committee. In particular, we introduce the so-called Vasicek one-factor model that was originally proposed in [6] for use in the forthcoming rules on capital charges.

We consider a portfolio loss variable L_n that is defined by

$$L_n = L_n(u_1, \dots, u_n) = \sum_{i=1}^n u_i \mathbf{1}_{\{\sqrt{\rho_i} X + \sqrt{1-\rho_i} \xi_i \leq c_i\}}, \quad (1)$$

where $u_i \geq 0$, $i = 1, \dots, n$, denotes the weight or the exposure of asset i in the portfolio, $0 < \rho_i < 1$ and $c_i \geq 0$, $i = 1, \dots, n$, are constants, and X, ξ_1, \dots, ξ_n are independent random variables with continuous distributions. The constants c_i are called *default thresholds*. They have to be calibrated in order to

fix the probabilities of default of the assets. The random variable X is interpreted as the change of an economic factor that influences all the assets in the portfolio but to different extents. The so-called *asset correlation* ρ_i measures the degree of the i -th asset's exposure to the systematic risk expressed by X . The random variables ξ_i are assumed to model the idiosyncratic (or specific) risk of the assets.

Equation (1) implies the following representation for the conditional variance of the loss L_n given X

$$\text{var}[L_n | X = x] = \sum_{i=1}^n u_i^2 \text{P}\left[\xi_i \leq \frac{c_i - \sqrt{\rho_i} x}{\sqrt{1 - \rho_i}}\right] \left(1 - \text{P}\left[\xi_i \leq \frac{c_i - \sqrt{\rho_i} x}{\sqrt{1 - \rho_i}}\right]\right). \quad (2)$$

We assume that the probabilities of default are not too small and that the correlations with the economic factor are not too high, i.e. in precise terms that $\inf_i c_i > -\infty$ and $\sup_i \rho_i < 1$. Then from (2) it follows that in case of independent, identically distributed ξ_1, ξ_2, \dots we have

$$\lim_{n \rightarrow \infty} \text{E}[\text{var}[L_n | X]] = 0 \quad \text{if and only if}^1 \quad \lim_{n \rightarrow \infty} \sum_{i=1}^n u_i^2 = 0. \quad (3)$$

Since

$$\text{var}[L_n] = \text{E}[\text{var}[L_n | X]] + \text{var}[\text{E}[L_n | X]], \quad (4)$$

the conditional expectation $\text{E}[L_n | X]$ appears to be a natural approximation of L_n as soon as (3) is fulfilled, i.e. as soon as the concentrations in the portfolio are not too big. Indeed, the approximation

$$L_n \approx \text{E}[L_n | X] \quad (5)$$

is fundamental for the Basel II approach to credit risk capital charges. For $\alpha \in (0, 1)$ and any random variable Y , define the α -quantile (or the Value-at-Risk (VaR)) of Y by

$$q_\alpha(Y) = \text{VaR}_\alpha(Y) = \inf\{y \in \mathbb{R} : \text{P}[Y \leq y] \geq \alpha\}. \quad (6)$$

Note that

$$\text{E}[L_n | X = x] = \sum_{i=1}^n u_i \text{P}\left[\xi_i \leq \frac{c_i - \sqrt{\rho_i} x}{\sqrt{1 - \rho_i}}\right]. \quad (7)$$

Since the right-hand side of (7) is a decreasing function in x , one then deduces from (5) that

$$q_\alpha(L_n) \approx \sum_{i=1}^n u_i \text{P}\left[\xi_i \leq \frac{c_i - \sqrt{\rho_i} q_{1-\alpha}(X)}{\sqrt{1 - \rho_i}}\right]. \quad (8a)$$

¹ Of course, here we admit an additional dependence of u_i on n , i.e. $u_i = u_{i,n}$.

Assuming that the ξ_i are all standard normally distributed then yields

$$q_\alpha(L_n) \approx \sum_{i=1}^n u_i \Phi\left(\frac{c_i - \sqrt{\rho_i} q_{1-\alpha}(X)}{\sqrt{1-\rho_i}}\right), \quad (8b)$$

where Φ denotes the standard normal distribution function. The linearity of the right-hand side of (8b) in the vector (u_1, \dots, u_n) suggests the choice of

$$\text{Basel II charge}(i) = u_i \Phi\left(\frac{c_i - \sqrt{\rho_i} q_{1-\alpha}(X)}{\sqrt{1-\rho_i}}\right) \quad (9)$$

as the capital requirement of asset i in the portfolio with the loss variable L_n . Up to an adjustment for the maturity of the loan which can be neglected in the context of this paper, (9) is just the form of the risk weight functions that was provided by the BCBS in [2].

3 Calculating Risk Contributions with the Semi-Asymptotic Approach

3.1 Definition of Semi-Asymptotic Capital Charges

In the following we review the approach by [5] for the definition of the risk contributions of single credit assets if risk is measured with Value-at-Risk (or just a quantile at fixed level). However, we consider here the risk measure Conditional Value-at-Risk which turned out to be more attractive from a conceptual point of view. We call our approach semi-asymptotic because, in contrast to Basel II where all exposures are assumed to be infinitely small, we keep one exposure fixed and let the others tend to infinitely small size.

We consider here a special case of (1) where $\rho_1 = \tau$, $c_1 = a$ but $\rho_i = \rho$ and $c_i = c$ for $i > 1$, and $\sum_{i=1}^n u_i = 1$. Additionally, we assume that $u_1 = u$ is a constant for all n but that u_2, u_3, \dots fulfills (3).

In this case, the portfolio loss can be represented by

$$L_n(u, u_2, \dots, u_n) = u \mathbf{1}_{\{\sqrt{\tau} X + \sqrt{1-\tau} \xi \leq a\}} + (1-u) \sum_{i=2}^n u_i \mathbf{1}_{\{\sqrt{\rho} X + \sqrt{1-\rho} \xi_i \leq c\}}, \quad (10)$$

with $\sum_{i=2}^n u_i = 1$. Transition to the limit for $n \rightarrow \infty$ in (10) leads to the *semi-asymptotic* percentage loss function

$$L(u) = u \mathbf{1}_D + (1-u) Y \quad (11)$$

with $D = \{\sqrt{\tau} X + \sqrt{1-\tau} \xi \leq a\}$ and $Y = \mathbb{P}\left[\xi \leq \frac{c - \sqrt{\rho} x}{\sqrt{1-\rho}}\right] \Big|_{x=X}$. Of course, a natural choice for τ might be $\tau = \rho$, the mean portfolio asset correlation.

For $\alpha \in (0, 1)$ and any random variable Z , define the the Conditional Value-at-Risk (CVaR) (or Expected Shortfall, see [1]) at level α of Z by

$$\text{CVaR}_\alpha(Z) = E[Z \mid Z \geq q_\alpha(Z)]. \quad (12a)$$

As by (11) we have

$$\begin{aligned} \text{CVaR}_\alpha(L(u)) &= u P[D \mid L(u) \geq q_\alpha(L(u))] \\ &\quad + (1 - u) E[Y \mid L(u) \geq q_\alpha(L(u))], \end{aligned} \quad (12b)$$

the following definition is rather near at hand.

The quantity

$$u P[D \mid L(u) \geq q_\alpha(L(u))] \quad (13)$$

is called semi-asymptotic CVaR capital charge (at level α) of the loan with exposure u (as percentage of total portfolio exposure) and default event D as in (11).

The capital charges we suggest in Definition (13) have to be calculated separately, i.e. for each asset an own model of type (11) has to be regarded. This corresponds to a bottom-up approach since the total capital requirement for the portfolio is determined by adding up all the capital charges of the assets. Note that the capital charges of Definition (13) are not portfolio invariant in the sense of [6]. However, in contrast to the portfolio invariant charges, the semi-asymptotic charges take into account not only correlation but also concentration effects. In particular, their dependence on the exposure u is not merely linear since also the factor $P[D \mid L(u) \geq q_\alpha(L(u))]$ depends upon u . Definition (13) is in line with the general definition of risk contributions (cf. [8], [11]) since (11) can be considered a two-assets portfolio model.

3.2 Calculation of Semi-Asymptotic Capital Charges

If F_0 and F_1 denote the conditional distribution functions of Y given $\mathbf{1}_D = 0$ and $\mathbf{1}_D = 1$ respectively, the distribution function of $L(u)$ is given by

$$P[L(u) \leq z] = p F_1\left(\frac{z-u}{1-u}\right) + (1-p) F_0\left(\frac{z}{1-u}\right), \quad (14)$$

where $p = P[D]$ is the *default probability* of the loan under consideration. By means of (14), the quantile $q_\alpha(L(u))$ can be numerically computed. For the conditional probability which is part of Definition (13), we obtain

$$P[D \mid L(u) \geq z] = \frac{p(1 - F_1(\frac{z-u}{1-u}))}{P[L(u) \geq z]}. \quad (15)$$

Denote by $\Phi_2(\cdot, \cdot; \theta)$ the distribution function of the bivariate standard normal distribution with correlation θ . If we assume that X and ξ are independent

and both standard normally distributed, we obtain $p = \Phi(a)$, and can derive for the conditional distribution functions from (14) that

$$F_1(z) = \begin{cases} 1 - p^{-1} \Phi_2\left(a, \frac{c - \sqrt{1-\rho} \Phi^{-1}(z)}{\sqrt{\rho}}; \sqrt{\tau}\right), & z \in (0, 1) \\ 0, & \text{otherwise,} \end{cases} \quad (16a)$$

and

$$F_0(z) = \begin{cases} (1 - p)^{-1} \Phi_2\left(-a, -\frac{c - \sqrt{1-\rho} \Phi^{-1}(z)}{\sqrt{\rho}}; \sqrt{\tau}\right), & z \in (0, 1) \\ 0, & \text{otherwise.} \end{cases} \quad (16b)$$

Let, similarly to the case of (16a) and (16b), $\Phi_3(\cdot, \cdot, \cdot; \Sigma)$ denote the distribution function of the tri-variate standard normal distribution with correlation matrix Σ . Define the function g by

$$g(x, \beta, z) = \frac{x - \sqrt{1-\beta} \Phi^{-1}(z)}{\sqrt{\beta}} \quad (17a)$$

and the correlation matrix $\Sigma_{\rho, \tau}$ by

$$\Sigma_{\rho, \tau} = \begin{pmatrix} 1 & \sqrt{\rho\tau} & \sqrt{\rho} \\ \sqrt{\rho\tau} & 1 & \sqrt{\tau} \\ \sqrt{\rho} & \sqrt{\tau} & 1 \end{pmatrix}. \quad (17b)$$

Then, in order to arrive at $\text{CVaR}_\alpha(L(u))$, the conditional expectation of Y given $\{L(u) \geq z\}$ from (12b) can be calculated according to

$$\begin{aligned} \mathbb{E}[Y | L(u) \geq z] &= \mathbb{P}[L(u) \geq z]^{-1} \left(\Phi_3\left(c, a, g(c, \rho, \frac{z-u}{1-u}); \Sigma_{\rho, \tau}\right) + \right. \\ &\quad \left. \Phi_2\left(c, g(c, \rho, \frac{z}{1-u}); \sqrt{\rho}\right) - \Phi_3\left(c, a, g(c, \rho, \frac{z}{1-u}); \Sigma_{\rho, \tau}\right) \right). \end{aligned} \quad (17c)$$

4 Numerical Example

We illustrate the previous results by a numerical example. In our focus is a portfolio that is driven by systematic risk only (the variable Y in (11)) and enlarge this portfolio with an additional loan (the indicator $\mathbf{1}_D$ in (11)).

In our example, the portfolio modeled by Y has a quite moderate credit standing which is expressed by its expected loss $\mathbb{E}[Y] = 0.025 = \Phi(c)$. By choosing $\rho = 0.1$ as asset correlation we arrive at a portfolio with a rather strong exposure to systematic risk. For the sake of simplicity we choose $\tau = \rho$, i.e. the exposure to systematic risk of the additional loan is identical to the exposure of the existing portfolio. However, we assume that the additional loan enjoys a quite high credit-worthiness as we set $p = \mathbb{P}[D] = 0.002 = \Phi(a)$.

Figure 1 illustrates the relative contribution of the new loan to the risk of

Risk contribution of new loan: different approaches

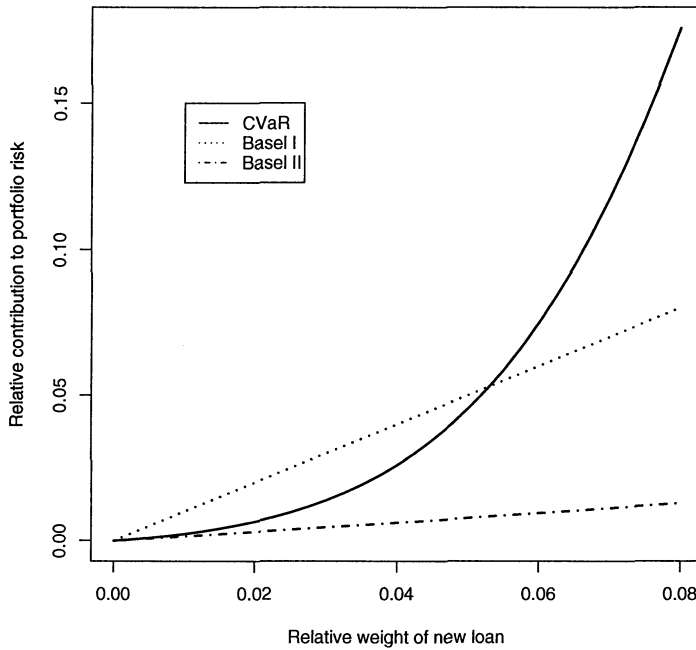


Fig.1. Relative risk contribution of new loan as function of the relative weight of the new loan. Comparison of contribution to true Conditional Value-at-Risk (CVaR), and the contributions according to the Basel II and Basel I Accords.

the portfolio loss variable $L(u)$. The contribution is expressed as a function of the relative weight u of the new loan in the portfolio and calculated according to three different methods. The first of the depicted methods relates to the relative contribution to true portfolio CVaR at level $\alpha = 99.9\%$, defined as the ratio of the contribution to CVaR according to Definition (13) and portfolio CVaR, i.e. the function

$$u \mapsto \frac{u P[D | L(u) \geq q_{\alpha}(L(u))]}{\text{CVaR}_{\alpha}(L(u))}, \quad (18)$$

where the conditional probability has to be evaluated by means of (15) and (16a). In the denominator, CVaR is calculated according to (12b) and (17c). Moreover, curves are drawn for the Basel II approach, i.e. the function

$$u \mapsto \frac{u \Phi\left(\frac{a - \sqrt{\tau} q_{1-\alpha}(X)}{\sqrt{1-\tau}}\right)}{u \Phi\left(\frac{a - \sqrt{\tau} q_{1-\alpha}(X)}{\sqrt{1-\tau}}\right) + (1-u) \Phi\left(\frac{c - \sqrt{\rho} q_{1-\alpha}(X)}{\sqrt{1-\rho}}\right)}, \quad (19)$$

and the Basel I approach. The latter approach just entails the diagonal as risk contribution curve since it corresponds to purely volume-oriented capital allocation.

Note that in Figure 1 the true CVaR curve intersects the diagonal (Basel I curve) just at the relative weight u^* that corresponds to the minimum risk portfolio $L(u^*)$. The Basel II curve differs strongly from the true contribution curve and is completely situated below the diagonal. This fact could yield the misleading impression that an arbitrarily high exposure to the additional loan still improves the risk of the portfolio. However, as the true CVaR curve in Figure 1 shows, the diversification effect from pooling with the new loan stops at 5.8% relative weight.

To sum up, it can be said that the example shows a shortcoming of the new Basel II capital requirement rules as they are not sensitive to concentrations. In addition, the example presents an intuitive bottom-up approach for calculating contributions that is sensitive to correlation as well as to concentrations and avoids time-consuming simulations.

References

1. ACERBI, C., TASCHE, D. (2002) Expected Shortfall: a natural coherent alternative to Value at Risk. *Economic Notes* 31(2), 379-388.
2. BASEL COMMITTEE ON BANKING SUPERVISION (2003A) The New Basel Capital Accord. Third consultative document.
3. BASEL COMMITTEE ON BANKING SUPERVISION (2003B) Trends in Risk Integration and Aggregation. Bank for International Settlements (ed.), Basel.
4. DENAULT, M. (2001) Coherent allocation of risk capital. *Journal of Risk* 4, no. 1, 1-34.
5. EMMER, S., TASCHE, D. (2003) Calculating credit risk capital charges with the one-factor model. *Working paper*.
6. GORDY, M. (2003) A Risk-Factor Model Foundation for Ratings-Based Bank Capital Rules. *Journal of Financial Intermediation* 12(3), 199-232.
7. KALKBRENER, M. (2002) An axiomatic approach to capital allocation. *Technical document, Deutsche Bank AG*.
8. LITTELMAN, R. (1996) Hot SpotsTM and Hedges. *The Journal of Portfolio Management* 22, 52-75. Special issue.
9. PATRIK, G., BERNEGGER, S., RÜEGG, M. (1999) The use of risk adjusted capital to support business decision making. In: Casualty Actuarial Society (Ed.), Casualty Actuarial Society Forum, Spring 1999 Edition, Baltimore.
10. ROCKAFELLAR, R., URYASEV, S. (2002) Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7), 1443-1471.
11. TASCHE, D. (1999) Risk contributions and performance measurement. *Working paper, Technische Universität München*.
12. THEILER, U. (2004) Risk-Return Management Approach for the Bank Portfolio. In: Szego, G. (ed.), Risk Measures for the 21st century, Wiley, pp. 403-431.

Integration der Simulation in die Programmplanung einer globalen Supply Chain

Lars Dohse¹, Thomas Hanschke², Ingo Meents¹, Horst Zisgen¹

¹ IBM Deutschland GmbH, Hechtsheimer Straße 2, 55131 Mainz

² TU Clausthal, Institut für Mathematik, Erzstraße 1, 38678 Clausthal-Zellerfeld

Zusammenfassung Die fortwährende Anwendung von Simulationsmodellen im Geschäftsprozess der taktischen Produktionsplanung scheitert oft an der Schwierigkeit, die Modellparameter immer auf dem aktuellen Stand zu halten. Dies ist umso schwieriger, je größer die Modelle werden. Allerdings ist gerade der Einsatz solcher Simulationsmethoden bei komplexen Modellen umso wichtiger, da z.B. Auswirkungen von geänderten Produktionsraten und Produktportfolios nicht direkt erkennbar sind. In diesem Beitrag wird beschrieben, wie das auf Warteschlangennetzwerken basierende Simulationsmodell EPOS (Enterprise Production Planning and Optimization System) der IBM als integriertes Simulationsmodell in die Planungsprozesse der IBM im Bereich der Festplattenfertigung eingebettet wurde. Dabei werden die mathematischen Grundlagen, der entwickelte Geschäftsprozess sowie Erfahrungen aus der betrieblichen Praxis dargelegt.

1 Einleitung

Die Notwendigkeit des Einsatzes von Simulationsmethoden, insbesondere zur Analyse komplexer Fertigungssysteme, ist unbestritten. Doch gerade für komplexe Fertigungssysteme ist es sehr schwierig die Modellparameter immer aktuell zu halten. Dies ist aber wiederum die Voraussetzung, um kurzfristige Simulationsanfragen befriedigen zu können. Soll die Simulation in einen übergeordneten Planungsprozess eingebunden und fortwährend genutzt werden, ist ein aktueller Datensatz umso wichtiger. Neben der Aktualität der Modellparameter ist eine schnelle Antwortzeit des Simulationssystems für den fortwährenden Einsatz enorm wichtig.

In der vorliegenden Arbeit wird beschrieben, wie das Problem der zeitnahen Aktualisierung der Daten und der Rechengeschwindigkeit mittels des integrierten Simulationssystems EPOS (Enterprise Production Planning and Optimization System) in der IBM Festplattenproduktion gelöst wurde. Ausgangspunkt ist die übergeordnete Hauptprogrammplanung für die weltweiten Fertigungsstätten von Festplatten und Komponenten. Nach einer kurzen Erläuterung der Schwierigkeiten bei der Hauptprogrammplanung im Umfeld der Festplattenindustrie werden die sich daraus resultierenden Anforderungen an den globalen Planungsprozess beschrieben. Im folgenden wird dann die Rolle der Simulation in diesem Prozess mit den speziellen Herausforderungen erklärt. Anschließend wird die Integration des Simulationssystems EPOS in diesen Planungsprozess erläutert. Insbesondere wird dabei dargelegt, wie die nötige Datenaktualität sichergestellt werden konnte und wie mittels der Verwendung von Warteschlangenmethoden die notwendigen Antwortzeiten für die Simulation erreicht wurden.

2 Planungsumfeld

Die Fertigung von Festplatten zeichnet sich durch eine hohe Fertigungstiefe aus (Abb.1). Dabei fanden die Fertigung der sogenannten Technologie-Komponenten (Schreib-Lese-Kopf (Wafer) und Dünnfilmplatte (Disk)) und die abschließenden Montageprozesse (Actuator und Laufwerk) bei der IBM statt. Die Komponentenfertigung und die Endmontage war auf über 10 Standorte auf 3 Kontinenten verstreut. Andere Komponenten, wie Motor oder Kabel, wurden von Lieferanten bezogen.

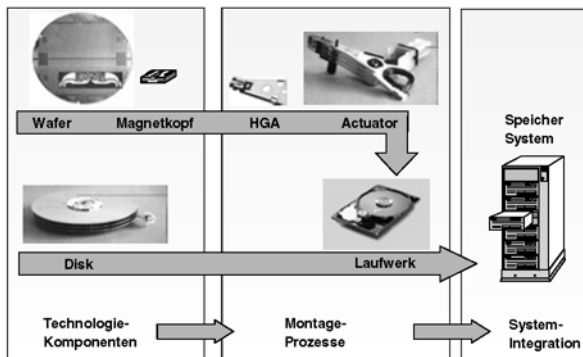


Abb. 1. Erzeugnisstruktur einer Festplatte

Ziel einer IBM divisionsweiten Initiative war es nun, ausgehend von den Absatzprognosen, eine zentrale Hauptprogrammplanung für die unterschiedlichen Komponenten und Fertigungsstätten unter Berücksichtigung der Kapazitäten an den verschiedenen Standorten, der aktuellen Lager- und Umlaufbestände an

Halbfabrikaten, sowie der Lieferzusagen der Lieferanten zu etablieren. Bei der Umsetzung galt es die für die Festplattenindustrie typischen Randbedingungen mit in Betracht zu ziehen. So ist eine Absatzprognose bei den sehr schnell schwankenden Nachfragen sehr schwierig. Auch sind die Produktlebenszyklen von sechs bis neun Monaten sehr kurz. Dabei ändert sich mit jedem neuen Produktanlauf die Kapazitätssituation aufgrund der technologischen Unterschiede. Damit ergibt sich eine starke Abhängigkeit der maximalen Fertigungskapazität vom jeweiligen Produktportfolio. Einen weiteren Einfluss auf die kapazitiven Nebenbedingungen haben die großen Variationen der Prozessparameter, wie z.B. die Prozessausbeute. Durch die unterschiedliche Komplexität der Komponentenfertigung ergeben sich auch große Unterschiede bei den zu planenden Vorlaufzeiten. Hinzu kommt, dass aufgrund der Vielzahl an Lieferanten eine Möglichkeit zur Lieferantenkollaboration geschaffen werden musste, damit verbindliche Lieferzusagen als Nebenbedingungen mit in die Planung einfließen können [1].

Es zeigte sich, dass über das eigentliche System zur Hauptprogrammplanung hinaus weitere Systeme zur Bereitstellung der notwendigen Planparameter eingeführt werden mussten. Die Abb. 2 skizziert die entsprechende Systemarchitektur, von der Hauptprogrammplanung über die Simulation von Kapazitäten und Vorlaufzeiten hin zur Lieferantenkollaboration. Im folgenden soll nun die Integration der Simulation in diesen Prozess beschrieben werden.

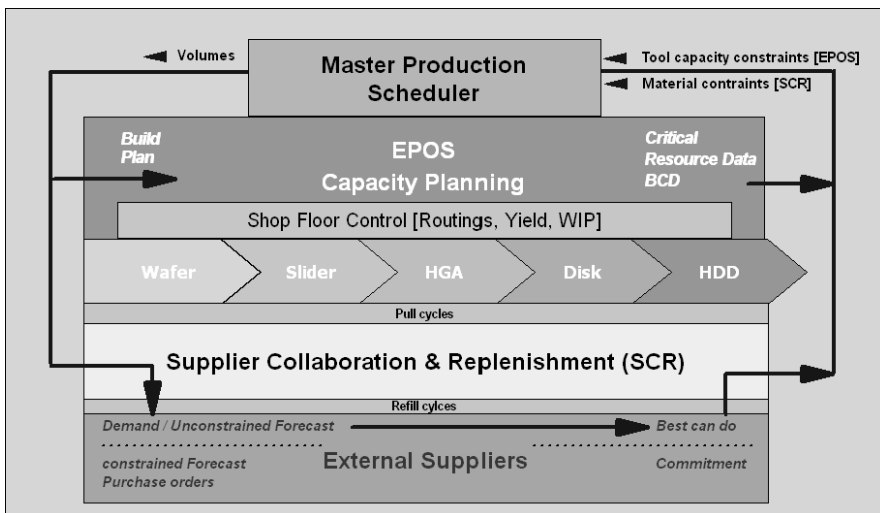


Abb. 2. Architektur der Planungssysteme

3 Integration der Simulation in die Hauptprogrammplanung

3.1 Prinzip der Integrierten Simulation

Die wesentlichen Nebenbedingungen der Hauptprogrammplanung sind die Kapazitäten an den Fertigungsstandorten. Die Bestimmung und Überprüfung dieser Nebenbedingungen ist aufgrund der Komplexität der Produktionslinien ohne ein entsprechendes Simulationsmodell nicht möglich. So hat z.B. alleine die Wafer-Fertigung pro Produkttyp mehr als 500 Arbeitsschritte, die auf ca. 300 verschiedenen Maschinen durchgeführt werden können. Hinzu kommt, dass die Wafer-Fertigung als typische Halbleiterproduktion mit vielen Schleifen extrem unübersichtlich ist, und auch von daher einer simulationsgestützten Planung bedarf. Um eine Einbindung der Simulationsergebnisse in die zentrale Hauptprogrammplanung zu erreichen, muss die Simulation als Geschäftsprozess in die Planungsabläufe integriert werden. Dabei lässt der strenge Zeitplan der Hauptprogrammplanung nur ein kurzes Zeitfenster von ein bis zwei Tagen für die Durchführung der Simulation und der endgültigen Zusage von Fertigungskapazitäten seitens des lokalen Fertigungsmanagements zu. Damit bleibt dem Planer keine Zeit, die Inputparameter des Simulationsmodells zu aktualisieren. Darüber hinaus muss die Antwortzeit des Simulationssystems sehr schnell sein, damit innerhalb des kurzen Zeitfensters die Durchführung von mehreren Simulationsläufen zur Bewertung von Alternativen überhaupt möglich ist. Letztendlich muss das Simulationssystem einen automatischen Austausch von Input- und Outputparametern mit der übergeordneten Hauptprogrammplanung erlauben.

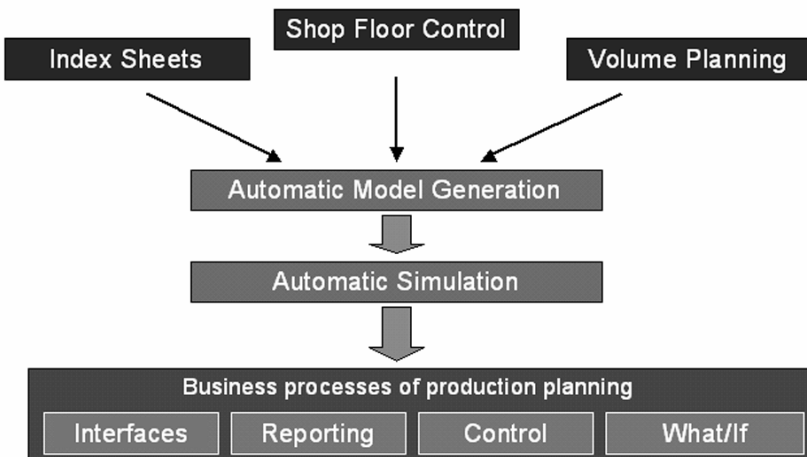


Abb. 3. Integrierte Simulation

3.2 Verteilte Dateneingabe

Fasst man allein die Wafer-Fertigung und die Endmontage der europäischen Werke zusammen, so ergeben sich über 1200 Arbeitsfolgen - durchgeführt auf ungefähr 3500 Maschinen und zusammengefasst zu ca. 550 Workcentern. Bei ca. 30 Parametern pro Workcenter und ca. 15 Parametern pro Arbeitsfolge wird die zu handhabende Datenmenge sehr groß.

Ein Teil der Parameter liegt schon in Betriebsdatenerfassungssystemen vor und wird über die entsprechenden Schnittstellen eingelesen. Andere Parameter müssen über manuelle Eingaben dem System zugeführt werden, da sie nicht in Datenbanken verfügbar sind, z.B. Planparameter für zukünftige Produkte. Für diese Dateneingabe wurde ein Geschäftsprozess entwickelt und mit Unterstützung von EPOS umgesetzt. Vorab wurden die Daten nach den Verantwortungsbereichen Wartung, Prozess und Fertigung untergliedert. Der Geschäftsprozess sieht vor, dass alle Ingenieure aus diesen drei Bereichen, die für die Inputparameter verantwortlich sind, die Möglichkeit der Dateneingabe erhalten. Die Dateneingabe muss dann vom Ingenieur selbst und dem verantwortlichen Management elektronisch bestätigt werden. Dabei hat die Fertigung das Recht, Parameter der anderen Bereiche zurückzuweisen, übernimmt im Gegenzug aber abschließend die Verantwortung für die Gültigkeit des gesamten Parametersatzes. Realisiert wurde dieses Konzept mit den Systemkomponenten DB/2, Lotus Notes und JAVA [4].

3.3 Automatische Modellgenerierung

Auf Grundlage der Datenbasis sollen Simulationsmodelle automatisch generiert werden. Die Modellgenerierung vollzieht sich dabei über mehrere Stufen, wobei objektorientierte Methoden die Erstellung von komplexen Abbildungen der Produktionsanlagen auf Simulationskomponenten erlauben. Auf diese Weise kann letztendlich aus einer Ansammlung von Betriebs- und Planungsdaten ein Simulationsmodell erstellt werden, dessen Ergebnisse wiederum im Kontext des Produktionsplaners interpretiert werden können. Die Transformation der von den Ingenieuren eingegebenen Parametern bis zur mathematischen Formel vollzieht sich in drei Schritten. Im ersten Schritt wird ein objekt-orientiertes Modell aus der Datenbank heraus erzeugt. Im zweiten Schritt wird dieses Modell auf ein äquivalentes Warteschlangennetzwerk abgebildet, wobei einzelne Arbeitsstationen durch Subnetzwerke ersetzt werden können. Der letzte Schritt ist die Generierung des mathematischen Modells.

3.4 Analytische Leistungsbewertung

Bei der Größe der zu simulierenden Fertigungslinien sind Simulationsmodelle auf der Basis der diskreten ereignisorientierten stochastischen Simulation zu rechenintensiv. Daher wurden Warteschlangennetzwerke aufbauend auf der Dekompositionsmethode entwickelt, wie sie in [2], [5] bzw. [6] zu finden ist. Die Knoten des Netzwerkes stellen allgemeine Mehrbedienersysteme mit Gruppenankünften und Gruppenbedienung der Form $G^X/G(b,b)/c$ dar, die mit Methoden der Diffusionsapproximation in geschlossener Form gelöst werden [3], [7]. Mit Hilfe der Warteschlangenmodelle werden die wichtigen Leistungsgrößen, wie die Maschinenauslastung, der maximale Durchsatz, die Nacharbeitsraten und die Prozessausbeute sowie die Durchlaufzeiten und die Umlaufbestände errechnet. Zur Prognose des Arbeitsfortschrittes wurde ein Fluss-Modell entwickelt, das auf dem Warteschlangenmodell aufbaut.

3.5 Web-Reporting und Schnittstellen

Ein zentraler Gedanke der *Integrierten Simulation* ist der Online-Zugriff auf alle Inputdaten und Simulationsergebnisse. Hierzu wurde ein dynamisches Web-Reporting entwickelt, das auf der Basis der in der Datenbank gespeicherten Simulationsergebnisse Berichte im HTML-Format erzeugt und über das Intranet zur Verfügung stellt. Für die Weiterverarbeitung der Simulationsergebnisse wurde auf der Basis von MQ-Series eine Schnittstelle zum Hauptprogrammplanungssystem etabliert. Dabei erhält der Planer über eine Filterfunktion die Möglichkeit, bestimmte Parameter auszublenden.

4 Fazit

Die Erfahrungen aus diesem Projekt zeigen, dass eine Integration der Simulation in regelmäßige Planungsabläufe mit einer verteilten Dateneingabe und einem zugrundeliegenden Geschäftsprozess möglich ist. So kann die notwendige Aktualität der Inputparameter von Simulationsmodellen, die die Fertigungskette eines Unternehmens modellieren, sichergestellt werden.

Der Online-Zugriff auf die Modellparameter wie auf die Simulationsergebnisse macht den Planungsprozess für alle Beteiligten transparent. Planung geschieht auf diese Weise nicht mehr isoliert in den Planungsabteilungen. Im Gegenteil, auch diejenigen Abteilungen, die vornehmlich für die Eingabe der Inputparameter verantwortlich sind, können die Planungsergebnisse einsehen und werden somit in den Planungsprozess eingebunden. Durch diese Durchgängigkeit wird das Verständnis für die Notwendigkeit einer akkuraten Dateneingabe und das Vertrauen in die Planungsergebnisse gefördert.

Eine weitere Herausforderung stellt die zentrale Hauptprogrammplanung für die weltweit verteilten und unterschiedlich komplexen Fertigungsstufen unter Einbeziehung der Lieferanten dar. Auch hier ist die Verfügbarkeit von aktuellen und akkuraten Inputdaten essentiell. Hier wurde durch die Verbindung mit der integrierten Simulation die Konsistenz der Planparameter durch den Zugriff auf eine zentrale Planparameter-Datenbank hergestellt. Mit dem Einsatz eines einheitlichen Datenmodells werden für alle Fertigungsabschnitte der Supply Chain untereinander vergleichbare Inputdaten für die übergeordnete Hauptprogrammplanung erzeugt und systemgestützt übermittelt.

Ein weiterer Vorteil der Kombination der Hauptprogrammplanung mit der Simulation ist die Möglichkeit zur Kontrolle der verwendeten Vorlaufzeiten und kapazitiven Nebenbedingungen im Simulationsmodell. Über diese Kontrolle, die dezentral in den verstreuten Fertigungsstätten durchgeführt wird, erfolgt dann die Zusage des Fertigungsmanagements, den vorgegebenen Plan zu erfüllen. Durch die engen Zeitvorgaben im Planungsprozess müssen die Simulationsergebnisse in recht kurzer Zeit vorliegen. Die entwickelten Warteschlangenmodelle eignen sich dafür in besonderer Weise, da alle im Zusammenhang mit der taktischen Planung benötigten Leistungsgrößen in sehr kurzer Zeit berechnet werden. Es bleibt dann noch genügend Zeit, um verschiedene Simulationsszenarien durchzutesten. Außerdem können durch die Weitergabe allein der kritischen Engpässen an die Programmplanung redundante Nebenbedingungen des Optimierungsproblems eliminiert werden.

Literatur

1. Fleck Th, Fromm H-J (2003) Supply Chain Management in der Praxis – Kollaborative Planung und Replenishment bei IBM. In: Dück O (Hrsg) Materialwirtschaft und Logistik in der Praxis. Weka Verlag
2. Gelenbe E (1975) On approximate computer system models. J of the ACM 22:261-269
3. Hanschke Th (1999) IBM-interner Bericht
4. Kramer M, Meents I (2001) Integrated Simulation – Optimization Strategies and Logistical Process Control for Production Planning based on Collaboratively Maintained Queueing Models. Dissertation, TU Clausthal
5. Pujolle G, Ai W (1986) A solution for multiserver and multiclass open queueing networks. INFOR 24:221-230
6. Whitt W (1983) The queueing network analyser. Bell System Technical Journal 62:2779-2825
7. Zisgen H (1999) Warteschlangennetzwerke mit Gruppenbedienung. Dissertation, TU Clausthal

Objektorientierte Simulation von Anlagen der Elektronikmontage

Grunow, M., Günther, H.-O.

Fachgebiet Produktionsmanagement, Technische Universität Berlin, Wilmersdorfer Str. 148, D-10585 Berlin, Email: {m.grunow, ho.guenther}@pm-berlin.net

Zusammenfassung Für die Leistungsanalyse hochautomatisierter Anlagen zur Montage elektronischer Baugruppen wurde ein objektorientiertes Simulationssystem entwickelt, das die zur Bestückung einer Leiterplatte erforderlichen Abläufe detailliert unter Berücksichtigung der kinematischen Eigenschaften der jeweiligen Bestückungsautomaten abbildet. Die entwickelten Simulationsbausteine sind für generelle Automatenklassen konzipiert und können daher durch einfache Parametrisierungen speziellen Bestückungsautomaten angepasst werden.

1. Einleitung

Bei der Herstellung elektronischer Baugruppen dienen Leiterplatten (PCBs, printed circuit boards) als Träger für die einzelnen elektronischen Bauelemente. Während früher bedrahtete Bauelemente verwendet wurden, die in manueller bzw. halbautomatischer Arbeitsweise bestückt wurden (siehe [9]), haben sich inzwischen oberflächenmontierte Bauelemente, sog. surface mount devices (SMDs) durchgesetzt, die mit Hilfe von Bestückungsautomaten direkt auf der Oberfläche der Leiterplatte aufgesetzt werden. Das Leistungsverhalten von einzelnen Bestückungsautomaten oder ganzen Montagelinien kann nur unzulänglich durch Faktoranalysen, Kennlinien oder einfache analytische Modelle erfasst werden. Um eine korrekte Abbildung des Leistungsverhaltens unter verschiedenen Rahmenbedingungen und Szenarien zu gewährleisten, ist eine detaillierte Simulation der Maschinenkinematik der Automaten in der Montagelinie erforderlich.

Am Fachgebiet Produktionsmanagement der TU Berlin wurde mit Unterstützung der Deutschen Forschungsgemeinschaft das simulationsgestützte Planungs- und Analysewerkzeug E A S E (Electronics Assembly Simulation Environment) entwickelt (siehe [1], [2], [5], [8]). Geeignet ist das System sowohl für die Auswahl und anwenderspezifische Modifikation von Bestückungsautomaten als auch für die operative Produktionsplanung und -steuerung. Die Simulationsmodelle in E A S E zeichnen sich durch eine sehr hohe Modellierungstiefe aus. In E A S E werden alle relevanten Maschinenelemente (Bestückungskopf, Arbeitstisch, Bauelementemagazin usw.) einzeln in ihrem Aufbau und ihrer dynamischen Funktionsweise als Simulationsbausteine modelliert. Darüber hinaus können vollständige

Montagelinien aus einzelnen Maschinenmodulen zusammengesetzt und weitere Elemente wie Siebdruckmaschinen oder Lötöfen in das Modell aufgenommen werden.

2. Typen von Bestückungsautomaten

Moderne Bestückungsautomaten bestehen im wesentlichen aus drei kinematischen Elementen: dem Magazin, das die Zuführungen für die verschiedenen Bauelemententypen aufnimmt, dem Leiterplattentisch, in den die Leiterplatte während der Bestückung eingespannt ist, sowie einem Transfersystem, das die Bauelemente vom Magazin zur Leiterplatte transportiert und den eigentlichen Bestückungsvorgang vornimmt. Die in der Industrie eingesetzten Typen von Bestückungsautomaten unterscheiden sich wesentlich durch ihre Arbeitsweise. In Abb. 1 ist der schematische Aufbau der gebräuchlichsten Automatentypen dargestellt.

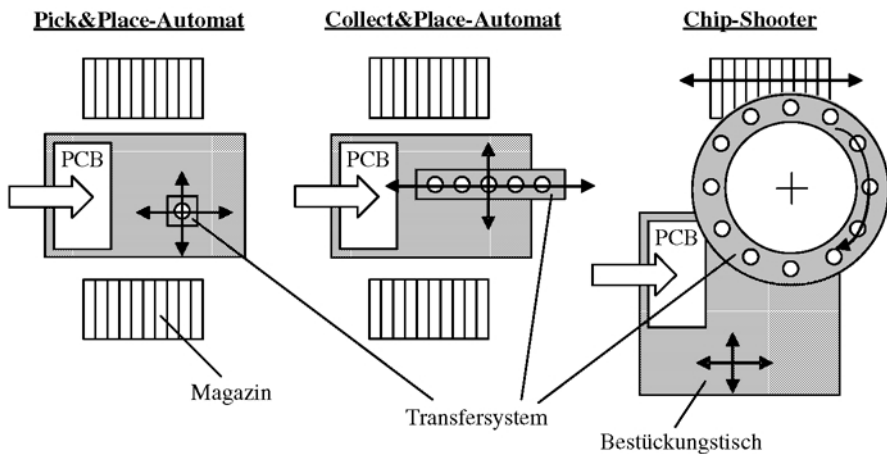
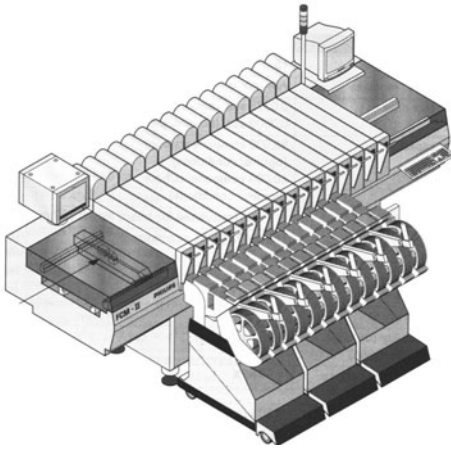


Abb. 1. Schematischer Aufbau verschiedener Bestückungsautomaten

Bei Automaten, die nach dem Pick&Place-Prinzip arbeiten, besteht das Transfersystem aus einem frei beweglichen Roboterarm, der ein einzelnes Bauelement aus dem Magazin aufnimmt und es anschließend auf der Leiterplatte aufsetzt. Eine Weiterentwicklung dieses Automatenprinzips stellt der Collect&Place-Automat dar, bei dem das Transfersystem als Revolver ausgelegt ist, der zuerst nacheinander mehrere Bauelemente aus dem Magazin aufnimmt und anschließend die geladenen Bauelemente an den vorbestimmten Positionen auf der Leiterplatte absetzt. Zentrale Optimierungsprobleme beim Einsatz von Collect&Place-Automaten stellen die Zuordnung der Bauelemententypen zu den Positionen im Magazin des Automaten sowie die Bildung von Bestückungstouren unter Beachtung der Kapazität des Revolvers dar (vgl. [7]). Einen dritten grundlegenden Automatentypen bilden die sog. Chip-Shooter, bei denen das Magazin in x-Richtung und der Bestü-

ckungstisch in x- und y-Richtung frei beweglich sind und das Transfersystem als Karussell ausgelegt ist, das durch schrittweise Drehbewegung die aufgenommenen Bauelemente weitertransportiert und sie schließlich auf der Leiterplatte absetzt. Die Besonderheit dieses Automatentyps ist die Positionierung von Magazin und Bestückungstisch simultan zur Drehbewegung des Transfersystems. Leistungsfähige Algorithmen für die Steuerung dieses Automatentyps finden sich in [4].

Die gestiegenen Produktivitätsanforderungen vor allem in der Massenproduktion von elektronischen Baugruppen haben in den letzten Jahren zu der Entwicklung von modularen Bestückungsautomaten geführt (siehe Abb. 2). Solche Auto-



maten bestehen aus einer Reihe von Pick&Place- oder Collect&Place-Modulen, die wie bei einer Transferstrasse sequentiell innerhalb des Automaten angeordnet sind. Die einzelnen Module sind durch ein Transportsystem (ähnlich einem Fließband) miteinander verbunden. Die Leiterplatten durchlaufen auf dem Transportsystem den Automaten und werden an den jeweiligen Modulen mit einer begrenzten Anzahl von Bauelementen bestückt.

Abb. 2. Modularer Bestückungsautomat

3. Optimierung modularer Bestückungsautomaten

Grundsätzlich bestehen bei der Konfiguration von Montagesystemen Wahlmöglichkeiten zwischen den einzelnen Automatentypen sowie hinsichtlich der Auslegung und Aufrüstung der jeweiligen Anlagen. Als Beispiel seien die zuvor erläuterten modularen Bestückungsautomaten betrachtet. Das zentrale Optimierungsproblem beim Einsatz dieses Automatentyps ist der Belastungsausgleich zwischen den einzelnen Modulen (vgl. [6]). Hierbei folgen wir der sog. unique-setup-Strategie, d.h. die Rüftung des Automaten erfolgt im Hinblick auf einen bestimmten Leiterplattentyp, der in großer Stückzahl hergestellt wird. Diese Rüststrategie steht im Gegensatz zur sog. minimum-setup-Strategie, die den Rüstzeitverlust zwischen unterschiedlichen, auf derselben Anlage gefertigten Leiterplattentypen minimiert (siehe [3], [8], [11]).

Die Leistungsabstimmung modularer Bestückungsautomaten gemäß der unique-setup-Strategie entspricht dem klassischen Problem der Fließbandabstimmung bei Einproduktproduktion, wobei die Module des Automaten den Arbeitsstationen und die einzelnen Bestückungsoperationen den Arbeitselementen der Fließproduk-

Entscheidungsvariablen

- x_{ijz} = 1, falls Bauelement i von Modul j im Arbeitstakt z bestückt wird
 (= 0, sonst)
 y_{jt} = 1, falls Bauelementtyp t im Magazin von Modul j aufgerüstet wird
 (= 0, sonst)
 w_z maximale Arbeitslast des Automaten im Arbeitstakt z

Modellformulierung

$$\min \sum_{z \in Z} w_z \quad (1)$$

unter den Nebenbedingungen

$$w_z \geq \sum_{i \in I_{jz}} x_{ijz} \quad j \in J, z \in Z \quad (2)$$

$$\sum_{t \in T} y_{jt} \leq m \quad j \in J \quad (3)$$

$$x_{ijz} \leq y_{j,t(i)} \quad j \in J, z \in Z, i \in I_{jz} \quad (4)$$

$$\sum_{z \in Z} \sum_{j \in J : i \in I_{jz}} x_{ijz} = 1 \quad i \in I \quad (5)$$

$$x_{ijz} \in \{0,1\} \quad j \in J, z \in Z, i \in I_{jz} \quad (6)$$

$$y_{jt} \in \{0,1\} \quad j \in J, t \in T \quad (7)$$

$$w_z \geq 0 \quad z \in Z \quad (8)$$

Die Zielfunktion (1) minimiert in Verbindung mit den Nebenbedingungen (2) die gesamte Arbeitslast des Automaten an den einzelnen Modulen. Die beschränkten Magazinkapazitäten werden in (3) erfasst. Aufgrund der Nebenbedingungen (4) ist eine Bestückungsoperation nur dann erlaubt, wenn auch das betreffende Bauelement in dem jeweiligen Modul gerüstet ist. Die Nebenbedingungen (5) stellen sicher, dass jede Bestückungsoperation ausgeführt wird. Hinzu kommen die Variablenbedingungen (6) bis (8).

Zur Lösung dieses komplexen Optimierungsproblems schlagen Grunow et al. [6] einen hierarchischen Ansatz vor, bei dem in der ersten Stufe mit Hilfe einer Heuristik die Zuordnung der Bauelementtypen zu den Modulen unter Beachtung der beschränkten Magazinkapazitäten erfolgt. In der zweiten Stufe wird dann alternativ mit Hilfe einer Heuristik bzw. eines einfachen binären Optimierungsmodells der Belastungsausgleich zwischen den Modulen durchgeführt. Wie Vergleiche mit einem einfach zu bestimmenden Lower Bound zeigen, wird in einer Vielzahl von Fällen die optimale Lösung gefunden.

4. Simulationsmodellierung

Die Simulation bietet die Möglichkeit, die Abläufe innerhalb eines Bestückungsautomaten wesentlich genauer als in dem obigen Optimierungsmodell abzubilden und darüber hinaus weitere Elemente des Montagesystems und der Anwendungsumgebung in das Modell einzubeziehen. Hierzu wird auf das eingangs erwähnte objektorientierte Simulationssystem E A S E zurückgegriffen, das mit Hilfe der Simulationssoftware eM-Plant und unter Einbindung externer C- und VisualBasic-Programme entwickelt wurde. E A S E besteht aus drei Grundmodulen, nämlich einer Modellbank mit Simulationsbausteinen für verschiedene Klassen von Bestückungsautomaten, einer Methodenbank mit Algorithmen zur Produktionsplanung und zur Automatensteuerung sowie verschiedenen Schnittstellen für die Einbindung vorhandener Produktionsdaten. Im Folgenden wird beispielhaft die Simulationsmodellierung eines modularen Bestückungsautomaten erläutert.

Das Simulationsmodell ist in zwei Hierarchieebenen gegliedert (vgl. [12]). In der oberen wird das Zusammenspiel zwischen den wichtigsten Systemelementen des Modells gesteuert. In der unteren Ebene erfolgt die detaillierte Modellierung der einzelnen Systemelemente (u.a. Leiterplatte, Portalsystem, Bestückungsköpfe, Transportsystem), die mit Steuerungsregeln zum Ablauf der Bestückungsvorgänge hinterlegt sind. Abb. 4 zeigt die Objekte des Simulationsmodells im Überblick.

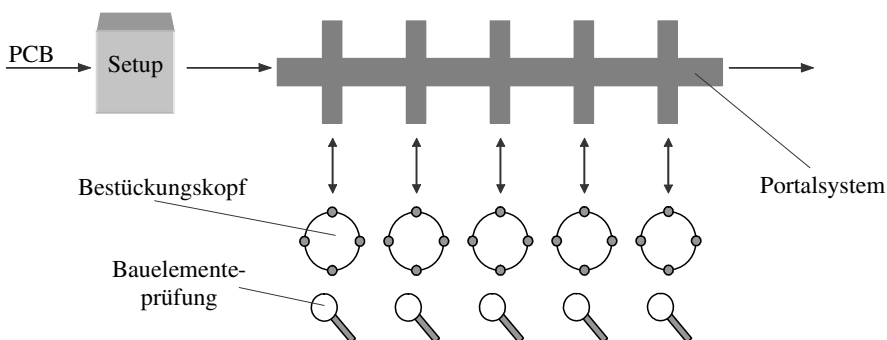


Abb. 4. Objekte des Simulationsmodells für einen modularen Bestückungsautomaten

In der Simulation werden sowohl die Hauptzeiten (Rüst- und Bestückungszeiten) als auch die Nebenzeiten (Leiterplattentransport, Ein- und Ausspannzeiten, Lagejustierung usw.) erfasst. Jede Leiterplatte durchläuft zunächst das Setup, in dem die Vorbereitung des Automaten bzw. der Leiterplatte zusammengefasst ist. Anschließend wird die Leiterplatte im Portalsystem des Automaten bestückt. Der Bestückungsvorgang wird für jedes Bauelement von der Entnahme aus dem Magazin, dem Transport zum Bestückungstisch, der Prüfung des Bauelements bis zur Platzierung auf der Leiterplatte vollständig abgebildet. Nach dem Bestücken sämtlicher Bauelemente verlässt die Leiterplatte das Portalsystem. Für eine spätere Auswertung werden die wichtigsten Daten der Simulation aufgezeichnet. Die Ein-

bindung externer Optimierungsprogramme (z.B. des im Vorabschnitt erwähnten Verfahrens zur Leistungsabstimmung) erfolgt über eine spezifische Schnittstelle.

Als modularer Bestückungsautomat wurde exemplarisch der Automat FCM 16 von Philips modelliert. Dieser Automat verfügt über 16 Module mit je einem Pick & Place-Bestückungskopf und einem Magazin für maximal sechs Bauelementtypen. Für die numerischen Untersuchungen wurden Daten von neun industriell gefertigten Leiterplatten verwendet, die in Autoradios und Navigationssystemen zum Einsatz kommen. Aus praktischer Sicht besteht ein wichtiges Anwendungsgebiet der Simulation darin, die reale Bestückungsleistung eines Automaten in Abhängigkeit von den Charakteristika der gefertigten Leiterplatten (u.a. Anzahl der Bauelemente sowie der Bauelementtypen) zu bestimmen. Abb. 5 zeigt als ausgewähltes Simulationsergebnis den Einfluss der Bauelementeanzahl auf die Bestückungsleistung eines modularen Bestückungsautomaten.

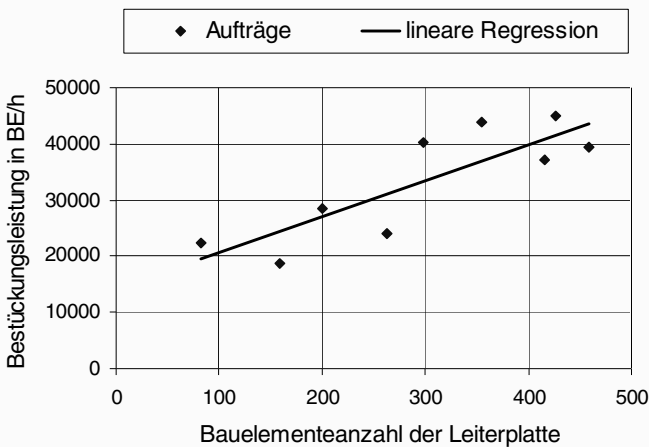


Abb. 5. Bestückungsleistung in Abhängigkeit von der Bauelementeanzahl

6. Fazit

Durch die Modellierung der Maschinenelemente sowie durch die Simulation jedes einzelnen Bestückungsvorgangs ist es möglich, ein verlässliches Abbild des realen Leistungsverhaltens des Produktionssystems unter Anwendung vorgegebener Steuerungsprogramme zu erhalten. Die objektorientierte modulare Programmierung bietet darüber hinaus die Möglichkeit, einzelne Systemelemente ohne größeren Aufwand zu manipulieren oder auszutauschen. Beispielsweise werden auf Automatenzebene Analysen der Automatenleistungen für konkrete Leiterplattentypen unterstützt. Dabei können auch einzelne Maschinenoperationen analysiert und Engpässe einzelner Maschinenelemente aufgezeigt werden. Auf diese Weise wird eine detaillierte Analyse des Leistungsverhaltens der Automaten ermöglicht.

Für den industriellen Betreiber eines Montagesystems erlaubt die Simulation eine konkrete Leistungsanalyse unter den Bedingungen der jeweiligen Anwendungsumgebungen, die u.a. durch die Produktionsorganisation, das Fertigungsprogramm und die Rüststrategien geprägt sind. Darüber hinaus können Entscheidungen zur Auswahl geeigneter Automatentypen und der Konfiguration des Montagesystems einschließlich der Kapazitätsdimensionierung und der Leistungsabstimmung innerhalb der Montagelinie unterstützt und alternative Verfahren zur Planung und Steuerung des Montagesystems und der einzelnen Automaten vergleichend bewertet werden.

Literatur

1. Föhrenbach A, Grunow M, Günther HO, Schleusener M (2000a) Leistungsabstimmung von Bestückungslinien – Einsatz eines modernen Simulationstools für die SMD-Montage. PLUS– Produktion von Leiterplatten und Systemen 2: 1609-1613
2. Föhrenbach A, Grunow M, Günther HO (2000b) Simulation und Optimierung von Montagesystemen für elektronische Baugruppen. Industrie-Management 16(3): 69-73
3. Gronalt M, Grunow M, Günther HO, Zeller R (1997) A heuristic for component switching on SMT placement machines. International Journal of Production Economics, 53: 181-190
4. Grunow M (2000) Optimierung von Bestückungsprozessen in der Elektronikmontage. Gabler – Deutscher Universitätsverlag, Wiesbaden
5. Grunow M, Günther HO, Föhrenbach A (2000) Simulation-based performance analysis and optimization of electronics assembly equipment. International Journal of Production Research 38: 4247-4259
6. Grunow M, Günther HO, Schleusener M (2003a) Component allocation for printed circuit board assembly using modular placement machines. International Journal of Production Research, 41: 1311-1331
7. Grunow M, Günther HO, Schleusener M, Yilmaz IO (2003b) Optimizing operations of a collect-and-place machine in PCB assembly. Working paper, TU Berlin
8. Günther HO, Föhrenbach A, Grunow M (1998) Simulation von Bestückungsautomaten in der Elektronikmontage. Zwf – Zeitschrift für wirtschaftlichen Fabrikbetrieb 93: 255-258
9. Günther HO, Gronalt M, Piller F (1996) Component kitting in semi-automated printed circuit board assembly. International Journal of Production Economics 43: 213-226
10. Günther HO, Gronalt M, Zeller R (1998) Job sequencing and component setup on a surface mount placement machine. Production Planning and Control 9: 201-211
11. Günther HO, Grunow M, Schorling Ch (1997) Workload planning in small lot printed circuit board assembly. OR Spektrum 19: 147-157
12. Nitschke S (2000) Simulation und Analyse von modularen Bestückungsautomaten in der Elektronikmontage. Diplomarbeit, Fachgebiet Produktionsmanagement, TU Berlin

A Mixed-Discrete Bilevel Programming Problem

Stephan Dempe¹ and Vyacheslav Kalashnikov²

¹ TU Bergakademie Freiberg, Freiberg, Germany

² Instituto de Tecnologías y Educación Superior de Monterrey, Monterrey, Nuevo León, 64849 México

Abstract. We investigate a special mixed-discrete bilevel programming problem resulting from the task to compute cash-out penalties due to the supply of incorrect amounts of natural gas through large gas networks. For easier solution we move the discreteness demand from the lower to the upper levels. This clearly changes the problem and it is our aim to investigate the relations between both problems. After the move, a parametric generalized transportation problem arises in the lower level for which the formulation of conditions for the existence of an optimal solution is our first task. In the second part of the paper, requirements guaranteeing that an optimal solution of the original problem is obtained by solving the modified one as well as bounds for the optimal value of the original problem are derived.

1 Introduction

Bilevel programming problems are hierarchical optimization problems where the set of variables is partitioned between two variables x and y . The values of $x = x(y)$ are determined by the *follower* knowing the selection of the other decision maker, the *leader*. The bilevel problem is the problem of the leader who has to determine the best choice of y knowing the reaction $x = x(y)$ of the follower on his decisions. To formulate the bilevel optimization problem consider the follower's parametric problem first. Given $y \in Y \subseteq \mathbb{R}^m$ he has to take

$$x(y) \in \Psi(y) := \underset{x}{\operatorname{Argmin}} \{f(x, y) : g(x, y) \leq 0\}, \quad (1)$$

i.e. $x(y)$ is an optimal solution of a parametric optimization problem. Now, the leader's problem (or the bilevel optimization problem) is

$$\min_{x, y} \{F(x, y) : x \in \Psi(y), y \in Y\}. \quad (2)$$

Problems of this type have been investigated in the monographs [1,2]. The number of possible applications is large and quickly growing (cf. the annotated bibliography [3]). Other than for continuous problems, (mixed-) discrete bilevel programming has been the topic of rather a small number of papers.

Here we consider a special bilevel programming problem where the lower level problem (1) is a generalized transportation problem:

$$\begin{aligned}
 & \left| \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n d_i^1 x_{i,m+1} + \sum_{j=1}^m d_j^2 x_{n+1,j} \right| \rightarrow \min_{x,q} \\
 & \sum_{j=1}^m x_{ij} + (1-q)x_{i,m+1} = a_i, \quad i = 1, \dots, n, \\
 & \sum_{i=1}^n u_{ij} x_{ij} + q u_{n+1,j} x_{n+1,j} = b_j, \quad j = 1, \dots, m, \\
 & x_{ij} \geq 0, \quad i = 1, \dots, n+1, \quad j = 1, 2, \dots, m+1, \\
 & q \in \{0, 1\}.
 \end{aligned} \tag{3}$$

Here, the coefficients $u_{ij} \in (0, 1)$ and the 0–1 Variable q indicates that we either have inequalities in the first or in the second block of constraints. This is a generalized transportation problem since the coefficients in the constraint matrix can be different from zero and one and possible slacks are penalized in the objective function. Moreover, the objective function is not linear but the absolute value of a linear function. Denote the set of optimal solutions of this problem by $\Psi(a, b)$, where $a = (a_1, \dots, a_n)^\top$, $b = (b_1, \dots, b_m)^\top$.

The bilevel programming problem reads then as

$$\begin{aligned}
 & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n d_i^1 x_{i,m+1} + \sum_{j=1}^m d_j^2 x_{n+1,j} \rightarrow \min_{x,q,a,b} \\
 & (x, q)^\top \in \Psi(a, b), \quad (a, b)^\top \in Y
 \end{aligned} \tag{4}$$

where Y is a fixed set, $Y \subseteq \mathbb{R}_+^n \times \mathbb{R}_+^m$.

Problem (3), (4) is a reformulation of a problem for minimizing the cash-out penalties of a natural gas shipping company [4,7,8].

Bilevel programming problems are \mathcal{NP} -hard [5], and discrete bilevel programming problems are even harder so solve. This is especially true if the discreteness demand is located in the lower level problem [9]. Hence, we feel that it may be advantageous to shift the integrality demand from the lower to the upper levels. This, of course, changes the problem, but it has two implications. First, since there is only one 0–1 variable in the problem, this is reduced to the solution of two bilevel programming problems, the better solution of the two is taken as result of the problem. Next, to solve the resulting two linear bilevel programming problems we can e.g. use an idea in [10] reducing both problems to quadratic optimization problems involving a penalty term in the objective function.

2 The Subproblems

If the discreteness demand is shifted into the upper level problem, the lower level ones (3) reduce to problems of minimizing the absolute value of a linear function on a polyhedron. If all the values on the right-hand side are finite numbers, the polyhedron is bounded. Hence, this problem has an optimal solution whenever its feasible set is nonempty. We give conditions guaranteeing nonemptiness of this set.

Consider the lower level problem (3) with $q = 1$ (the case with $q = 0$ can be dealt with similarly) and take an arbitrary linear objective function for convenience:

$$\begin{aligned}
 & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m d_j^2 x_{n+1,j} \rightarrow \min_{x,q} \\
 & \sum_{j=1}^m x_{ij} = a_i, \quad i = 1, \dots, n+1, \\
 & \sum_{i=1}^n u_{ij} x_{ij} + u_{n+1,j} x_{n+1,j} = b_j, \quad j = 1, \dots, m, \\
 & x_{ij} \geq 0, \quad i = 1, \dots, n+1, \quad j = 1, 2, \dots, m.
 \end{aligned} \tag{5}$$

Note that we have added one constraint with unknown (to be fixed later) but nonnegative right-hand side a_{n+1} to give the model a more familiar formulation. If this problem has a feasible solution, then this is also valid for the original one. Now, we use an idea originating from [6] to transform the problem. Take any set of $m + (n+1) - 1 = m + n$ basic variables x_{ij} , $(i, j) \in U$. Then, if we take $s_1 = 1$, the system of equations

$$s_i - t_j u_{ij} = 0, \quad \forall (i, j) \in U \tag{6}$$

has a unique solution (s, t) with $s_i > 0$, $t_j > 0$ for all i, j . Using this solution, the generalized transportation problem can be transformed into

$$\begin{aligned}
 & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m d_j^2 x_{n+1,j} \rightarrow \min_{x,q} \\
 & \sum_{j=1}^m s_i x_{ij} = s_i a_i, \quad i = 1, \dots, n+1, \\
 & \sum_{i=1}^n u_{ij} \frac{t_j}{s_i} s_i x_{ij} + u_{n+1,j} \frac{t_j}{s_{n+1}} s_{n+1} x_{n+1,j} = t_j b_j, \quad j = 1, \dots, m, \\
 & x_{ij} \geq 0, \quad i = 1, \dots, n+1, \quad j = 1, 2, \dots, m.
 \end{aligned} \tag{7}$$

This problem arises if each equation in the first set is multiplied by s_i and each one in the second system by t_j . But now, the coefficients in the second

set of equations reduce to one for $(i, j) \in U$. By simply summing up the constraints of problem (7) and using that $x_{ij} = 0$ for $(i, j) \notin U$ we get

Theorem 1. *If the generalized transportation problem (5) with $q = 1$ has a feasible solution then there is a basic feasible solution with $x_{ij} = 0$ for $(i, j) \notin U$ such that*

$$\sum_{i=1}^{n+1} s_i a_i = \sum_{j=1}^m t_j b_j \quad (8)$$

for the solution s, t of (6).

Equation (8) shows that a_{n+1} has to be taken as

$$s_{n+1} a_{n+1} = \sum_{j=1}^m t_j b_j - \sum_{i=1}^n s_i a_i$$

and that a necessary condition for solvability of this problem is $a_{n+1} \geq 0$. Not every set of basic variables corresponds to feasible solutions of the generalized transportation problem (5). To find a condition characterizing correct basic solutions, consider a subset $I \subseteq \{1, \dots, n+1\}$ and the set $N(I) := \{j : (i, j) \in U \text{ for some } i \in I\}$. Then, for a feasible solution of the generalized transportation problem we have

$$\sum_{i \in I} s_i a_i = \sum_{i \in I} \sum_{(i, j) \in U} s_i x_{ij} \leq \sum_{j \in N(I)} \sum_{(i, j) \in U} u_{ij} \frac{t_j}{s_i} s_i x_{ij} = \sum_{j \in N(I)} t_j b_j \quad (9)$$

due to $x_{ij} = 0$ for $(i, j) \notin U$. Here, the inequality is implied by $\{(i, j) \in U : i \in I\} \subseteq \{(i, j) \in U : j \in N(I)\}$ and $u_{ij} t_j / s_i = 1$ for $(i, j) \in U$. Also, using the set $M(J) := \{i : (i, j) \in U \text{ for some } j \in J\}$ for $J \subseteq \{1, \dots, m\}$ we get the similar inequality

$$\sum_{i \in M(J)} s_i a_i = \sum_{i \in M(J)} \sum_{(i, j) \in U} s_i x_{ij} \geq \sum_{j \in J} \sum_{(i, j) \in U} u_{ij} \frac{t_j}{s_i} s_i x_{ij} = \sum_{j \in J} t_j b_j. \quad (10)$$

Theorem 2. *Consider the generalized transportation problem (5) with $q = 1$. Then a set of basic variables corresponds to a feasible solution if and only if the conditions (8) together with (9) and (10) are satisfied for all subsets $I \subseteq \{1, \dots, n+1\}$ and $J \subseteq \{1, \dots, m\}$.*

To prove this theorem use a method like the north-west corner rule for computing a starting solution for the transportation problem. Such methods fix the basic variables one after the other such that in any step one of the equations in the definition of the problem (7) is satisfied. Then, in the next step, a basic variable x_{ij} is to be determined with

$$(i, j) \in \{(i, j) \in U : j \in N(I)\} \setminus \{(i, j) \in U : i \in I\}$$

respectively

$$(i, j) \in \{(i, j) \in U : i \in M(J)\} \setminus \{(i, j) \in U : j \in J\}$$

for which is enough space by the assumptions of the theorem. Here, U denotes the index set of basic variables, and I and J are the index sets of the equations being already satisfied in that step.

To derive another result consider the variables $x_{n+1,j}$ in the problem (5) as being slack variables. Then, to investigate solvability of the lower level problems we can investigate the two problems

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} &\rightarrow \min_{x,q} \\ \sum_{j=1}^m x_{ij} &= a_i, \quad i = 1, \dots, n, \\ \sum_{i=1}^n u_{ij} x_{ij} &\leq b_j, \quad j = 1, \dots, m, \\ x_{ij} &\geq 0, \quad i = 1, \dots, n, \quad j = 1, 2, \dots, m \end{aligned} \tag{11}$$

and

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} &\rightarrow \min_{x,q} \\ \sum_{j=1}^m x_{ij} &\leq a_i, \quad i = 1, \dots, n, \\ \sum_{i=1}^n u_{ij} x_{ij} &= b_j, \quad j = 1, \dots, m, \\ x_{ij} &\geq 0, \quad i = 1, \dots, n, \quad j = 1, 2, \dots, m. \end{aligned} \tag{12}$$

Theorem 3. For any natural numbers n, m and vectors $a = (a_1, \dots, a_n)^\top$, $b = (b_1, \dots, b_m)^\top$, with $a_i > 0, i = 1, \dots, n, b_j > 0, j = 1, \dots, m$ at least one of the problems (11) and (12) has a feasible solution.

Proof. To prove our assertion, we use the modified North-West Corner Rule in the following form.

0: Set $k = 0, i_k = 1, j_k = 1$.

1: Put

$$x_{i_k, j_k} = \begin{cases} a_{i_k}, & \text{if } u_{i_k, j_k} a_{i_k} \leq b_{j_k}; \\ b_{j_k} / u_{i_k, j_k}, & \text{otherwise.} \end{cases}$$

2: If $i_k < n$ then set

$$i_{k+1} = \begin{cases} i_k + 1, & \text{if } u_{i_k, j_k} a_{i_k} \leq b_{j_k}, \\ i_k, & \text{otherwise.} \end{cases}$$

If $i_k = n$ and $u_{i_k, j_k} a_{i_k} \leq b_{j_k}$, then set $K = k$ and stop. If $i_{k+1} = i_k + 1$ goto Step 3.

Symmetrically, if $j_k < m$, put

$$j_{k+1} = \begin{cases} j_k + 1, & \text{if } u_{i_k, j_k} a_{i_k} \geq b_{j_k}, \\ j_k, & \text{otherwise.} \end{cases}$$

At last, if $j_k = m$ and $u_{i_k, j_k} a_{i_k} \geq b_{j_k}$, then set $K = k$ and stop.

3: Now we update the parameters:

$$a_{i_k} := a_{i_k} - x_{i_k j_k}; \quad b_{j_k} := b_{j_k} - u_{i_k j_k} x_{i_k j_k}.$$

If $a_i = 0$ for all $i = 1, \dots, n$, or $b_j = 0$ for each $j = 1, \dots, m$, then set $K = k$ and stop. Otherwise, set $k := k + 1$ and return to Step 1.

Let the algorithm stop in Step 3. Then by construction, either $a = 0$ which means that we have $\sum_{j=1}^m x_{ij} = a_i$, $i = 1, \dots, n$ or $b = 0$ implying $\sum_{i=1}^n u_{ij} x_{ij} = b_j$, $j = 1, \dots, m$. Since the other system of inequalities is also satisfied, we obtain non-emptiness of the feasible set of either problem (11) or (12). If the algorithm stops in Step 2 then either $i_K = n$ and $\sum_{j=1}^m x_{ij} = a_i$, $i = 1, \dots, n$ or $j_K = m$ and $\sum_{i=1}^n u_{ij} x_{ij} = b_j$, $j = 1, \dots, m$. In both cases, the other system of inequalities is also satisfied by construction. This again implies non-emptiness of the feasible set of either problem (11) or (12). The theorem is proved completely.

3 Relation Between the Bilevel Problems

As mentioned in the Introduction, we replaced problem (3), (4) by

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n d_i^1 x_{i, m+1} + \sum_{j=1}^m d_j^2 x_{n+1, j} \rightarrow \min_{x, q, a, b} \\ (x, q)^\top \in \Psi_q(a, b), (a, b)^\top \in Y, q \in \{0, 1\} \end{aligned} \quad (13)$$

where $\Psi_q(a, b)$ denotes the set of optimal solutions of problem (3) with q being fixed to $q = 0$ or $q = 1$. In doing so we cannot expect to obtain optimal solutions for problem (3), (4) in any case.

Theorem 4. *Let (x^0, a^0, b^0) be a locally optimal solution for problem*

$$\min_{x, a, b} \left\{ \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n d_i^1 x_{i, m+1} : (x, q)^\top \in \Psi_q(a, b), (a, b)^\top \in Y \right\} \quad (14)$$

with $q = 0$ and objective function value

$$z^0 = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}^0 + \sum_{i=1}^n d_i^1 x_{i,m+1}^0$$

and let $x^1 \in \Psi_1(a^0, b^0)$ with

$$z^1 = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}^1 + \sum_{j=1}^m d_j^2 x_{n+1,j}^1.$$

If $0 < z^0 < z^1$, then (x^0, a^0, b^0) is a local optimal solution of problem (3), (4).

The proof of this theorem mainly uses continuity of the optimal function value of right-hand side perturbed linear optimization problems guaranteeing that the function value of feasible solutions for the subproblem with $q = 1$ for parameter values being near to (a^0, b^0) can never be lower than z^0 . Hence the sequence of inequalities $0 < z^0 < z^1$ implies that $x^0 \in \Psi(a^0, b^0)$ is feasible for the problem (3), (4) and this persists in a neighborhood of (a^0, b^0) .

There are some more cases which can be treated similarly:

- Remark 1.*
1. (x^0, a^0, b^0) is locally optimal for problem (14) with $q = 0$ and $0 > z^0 > z^1$,
 2. (x^1, a^1, b^1) is locally optimal for problem (14) with $q = 1$ and $0 > z^1 > z^0$, and
 3. (x^1, a^1, b^1) is again locally optimal for problem (14) with $q = 1$ and $0 < z^1 < z^0$.

It is not too difficult to see that an optimal solution of problem (3) is either in $\Psi_0(a, b)$ or in $\Psi_1(a, b)$. Hence,

$$\Psi(a, b) \subseteq \Psi_0(a, b) \cup \Psi_1(a, b)$$

and thus

$$\{(x, a, b) : x \in \Psi(a, b)\} \subseteq \{(x, a, b) : x \in \Psi_0(a, b)\} \cup \{(x, a, b) : x \in \Psi_1(a, b)\}.$$

Now, since the feasible set in problem (3), (4) is equal to the set on the left-hand side and the feasible set of the really solved one is equal to the larger set on the right-hand side of this inclusion we get

Theorem 5. $\min_{q \in \{0,1\}} z^q \leq z^* \leq \max_{q \in \{0,1\}} z^q$ where z^q denotes the optimal function value of (14) and z^* is the optimal value of problem (3), (4).

References

1. Bard, J. F. (1998) Practical Bilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers, Dordrecht
2. Dempe, S. (2002) Foundations of Bilevel programming. Kluwer Academic Publishers, Dordrecht
3. Dempe, S. (2003) Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints. Optimization 52, 333-359
4. Dempe, S., Kalashnikov, V. (2002) Discrete bilevel programming: application to a gas shipper's problem. Preprint No. 2002-02, TU Bergakademie Freiberg, (available at:

http://www.mathe.tu-freiberg.de/~dempe/Artikel/gas_cash-out.ps)
5. Deng, X. (1998) Complexity issues in bilevel linear programming. In: Migdalas, A., Pardalos, P., and Värbrand, P. (Eds.): Multilevel Optimization: Algorithms and Applications, , Kluwer Academic Publishers, Dordrecht et al., 149-164.
6. Gabasov, R.F., Kirillova, F.M. (1978) Linear Programming Methods. Part 2. Transportation Problems (Metody linejnogo programmirovaniya. Chast' 2. Transportnye zadachi). Izdatel'stvo Belorusskogo Universiteta, Minsk, (in Russian)
7. Kalashnikov, V.V., Ríos-Mercado, R.Z. (2001) A penalty-function approach to a mixed-integer bilevel programming problem. In: Zozaya, C. e.a. (Eds.), Proceedings of the 3rd International Meeting on Computer Science, Aguascalientes, Mexico, Vol. 2, pp. 1045-1054
8. Kalashnikov, V.V., Ríos-Mercado, R.Z. (2002) An algorithm to solve a gas cash-out problem. In: R. Clute (Ed.), Proceedings of the International Business and Economic Research Conference (IBERC'2002), Puerto Vallarta, Mexico
9. Vicente, L.N., Savard, G., Judice, J.J. (1996) The discrete linear bilevel programming problem. Journal of Optimization Theory and Applications 89, 597-614.
10. White, D.J., Anandalingam, G. (1993) A penalty function approach for solving bilevel linear programs. Journal of Global Optimization 3, 397-419

Detecting Superfluous Constraints in Quadratic Programming by Varying the Optimal Point of the Unrestricted Problem

Peter Recht

Universität Dortmund, Operations Research und Wirtschaftsinformatik,
Vogelpothsweg 87, D 44 221 Dortmund, Germany,

Abstract. In [3] QP -problems with strict convex objective functions f were investigated in order to detect constraints that are not active at the optimal point x^* . It turned out, that simple calculations performed in parallel with an QP -solver allow a decision to delete restrictions from the problem. This might shrink down the problem size during the optimization procedure. These sufficient conditions for non activity could be generalized to the positive semi-definite case (see [4]). In the present paper it is shown that the techniques presented in [3] allow to get additional possibilities for a deletion of superfluous constraints. For this, the optimal point x_0 of the unrestricted QP -problem is suitably disturbed within some special set $M(f(x^*))$. This set is unknown in advance but a subset $A(x_0)$ can algorithmically be generated. It is also proved, that x^* is a limit point in $A(x_0)$.

1 Introduction

We consider the strictly convex quadratic programming problem **QP**

$$\begin{aligned} \min_{x \in S} f(x) &= \frac{1}{2}(x - x_0)^t C(x - x_0) \\ \text{s.t. } S &= \{x \mid a_i^t x \leq b_i, i = 1, 2, \dots, m\}. \end{aligned}$$

C is a symmetric, positive definite $(n \times n)$ matrix and $d \in \mathbb{R}^n$. We assume that the set $S \subset \mathbb{R}^n$ of feasible points is non-empty and that all $a_i \neq 0$. Let x^* denote the optimal solution of **QP**.

We will say that the i -th restriction is *superfluous* for **QP**, if it is non-active at x^* , i.e. $a_i^t x^* < b_i$.

For an arbitrary point $y \in \mathbb{R}^n$ define the following structural quantities and sets that appear in connection with **QP**:

$$\begin{aligned} r_i(y) &:= a_i^t y - b_i, & i &= 1, 2, \dots, m \\ q_{ij} &:= a_i^t C^{-1} a_j, & i, j &= 1, 2, \dots, m \\ J_-(y) &:= \{i \mid r_i(y) < 0\} \\ J_+(y) &:= \{i \mid r_i(y) > 0\} \end{aligned}$$

The aim of the paper is, to use these quantities for detecting superfluous restrictions for **QP**.

2 Characterizing Active and Superfluous Constraints

Consider an arbitrary positive definite, symmetric $(n \times n)$ matrix \tilde{C} . Together with y and the a_i 's it induces quantities $\tilde{q}_{ij} = a_i^t \tilde{C}^{-1} a_j$ and a function $\tilde{F}(x, y) = \frac{1}{2}(x - y)^t \tilde{C}(x - y)$. They provide a complete characterization of superfluous constraints for **QP**:

Proposition 1. *The i -th restriction is superfluous for **QP** if and only if there is $y \in \mathbf{R}^n$, a positive definite, symmetric matrix \tilde{C} and $\alpha > 0$, satisfying the conditions*

- a.) $\tilde{F}(x^*, y) \leq \alpha$
- b.) $i \in J_-(y)$ and $r_i^2(y) > 2\alpha\tilde{q}_{ii}$.

Proof: " \Rightarrow ": Choose $y = x^*$, $\tilde{C} = I_n$, $\tilde{\alpha} < \frac{1}{2} \cdot (b_i - a_i^t x^*)^2 \|a_i\|^{-2}$. With these settings conditions a.) and b.) trivially hold.

" \Leftarrow ": Let $\tilde{x}^{(i)*}$ be the minimizer of $\tilde{F}(x, y)$ on the affine space $H_i := \{x \mid a_i^t x = b_i\}$.

Computation yields $\tilde{x}^{(i)*} = y + \lambda \tilde{q}_{ii}^{-\frac{1}{2}} \tilde{C}^{-1} a_i$, where $\lambda = -r_i(y) \tilde{q}_{ii}^{-\frac{1}{2}}$. Since $i \in J_-(y)$, $\lambda > 0$. Further on $F(\tilde{x}^{(i)*}, y) = \frac{1}{2} \lambda^2 = \frac{1}{2} r_i^2(y) \tilde{q}_{ii}^{-1}$. Hence, for an arbitrary point $x \in S$ with $\tilde{F}(x, y) \leq \alpha$ we get $\tilde{F}(x, y) < \tilde{F}(\tilde{x}^{(i)*}, y)$. For these points Taylor formula $\tilde{F}(x, y) = \tilde{F}(\tilde{x}^{(i)*}, y) + \nabla_x \tilde{F} \big|_{x=\tilde{x}^{(i)*}}^t (x - \tilde{x}^{(i)*}) + \frac{1}{2}(x - \tilde{x}^{(i)*})^t \tilde{C}(x - \tilde{x}^{(i)*})$ implies $\nabla_x \tilde{F} \big|_{x=\tilde{x}^{(i)*}}^t (x - \tilde{x}^{(i)*}) < 0$.

This leads to $\lambda \tilde{q}_{ii}^{-\frac{1}{2}} a_i^t (x - \tilde{x}^{(i)*}) < 0$, i.e. $a_i^t (x - \tilde{x}^{(i)*}) < 0$. Since $x \in S$ was arbitrarily chosen, strict inequality must hold also at x^* , i.e. $a_i^t x^* < b_i$. ■

For an algorithmic purpose only the " \Leftarrow " part is of interest but, unfortunately, the proposition is not constructive in this regard. It gives no hints how to get y , \tilde{C} and α . Moreover, x^* is unknown in advance.

In order to exploit it anyhow for the detection of superfluous restrictions we will fix \tilde{C} by the original matrix C but allow y and α to vary. The variation of y can be considered as alteration of the point x_0 , the optimal point of the unrestricted **QP**-problem. The special case, that $y = x_0$ is fixed, was investigated in [3].

For the values of α we allow a variation within the interval $[f(x^*), \infty[$, i.e. we will generally suppose that α is given in such a way that $f(x^*) \leq \alpha$. From a practical point of view, the knowledge of such α is not a crucial restriction. Standard **QP**-solution procedures that are of interior point type successively produce a decreasing sequence of such upper bounds (e.g. see [1]), [2].

As a function of y , $F(x^*, y) := \frac{1}{2}(x^* - y)^t C(x^* - y)$ induces convex level sets $M(\alpha) := \{y \mid F(x^*, y) \leq \alpha\}$ for $\alpha \geq 0$. These sets are monotonous in $\alpha \geq 0$ with respect to set-inclusion.

Note, that for $\alpha \geq f(x^*)$ $x_0 \in M(\alpha)$.

Superfluous restrictions can now be characterized by considering only points $y \in M(\alpha)$, i.e. varying x_0 within $M(\alpha)$.

Proposition 2. *The i -th restriction is superfluous for **QP** if and only if there is $y \in M(\alpha)$ satisfying $i \in J_-(y)$ and $r_i^2(y) > 2\alpha q_{ii}$.*

Proof: " \Rightarrow ". Let $r_i(x^*) < 0$. Then there is $x_i^* := x^* - r_i^2(x^*)q_{ii}^{-1}C^{-1}a_i$ s.t. $a_i^t x_i^* = b_i$. Obviously, $F(x^*, x_i^*) = \frac{r_i^2(x^*)}{2q_{ii}}$. Now, consider the case $F(x^*, x_i^*) \leq \alpha$. Define $y := x^* + \lambda q_{ii}^{-1}C^{-1}a_i$, where λ is arbitrarily taken from the interval $\left[r_i(x^*) \left(\frac{\alpha}{F(x^*, x_i^*)} \right)^{\frac{1}{2}}, r_i(x^*) \left[\left(\frac{\alpha}{F(x^*, x_i^*)} \right)^{\frac{1}{2}} - 1 \right] \right]$.

For such y we get $F(x^*, y) = \frac{1}{2}\lambda^2 q_{ii}^{-1} \leq \frac{1}{2}r_i^2(x^*) \left(\frac{\alpha}{F(x^*, x_i^*)} \right) q_{ii}^{-1} = \alpha$, i.e. $y \in M(\alpha)$. Moreover, $a_i^t y = a_i^t x^* + \lambda$ and $r_i(y) = r_i(x^*) + \lambda$. Since $\lambda < 0$, we get $i \in J_-(y)$ and $r_i^2(y) = (r_i + \lambda)^2 > (r_i(x^*) + r_i(x^*) \left(\frac{\alpha}{F(x^*, x_i^*)} \right)^{\frac{1}{2}} - 1)^2 = 2\alpha q_{ii}$. I.e. (ii.) is satisfied. This is also the case if $\alpha < F(x^*, x_i^*)$, by simply setting $y := x^*$.

" \Leftarrow ". This follows immediately by Prop. 1. ■

Since α is an arbitrary upper bound for the optimal value of **QP** we immediately have

Corollary 1. *The i -th restriction is superfluous for **QP** if and only if there is $y \in M(f(x^*))$ satisfying $i \in J_-(y)$ and $r_i^2(y) > 2f(x^*)q_{ii}$.*

We now also get a complete characterization of *active* constraints:

Proposition 3. *The i -th restriction is active at x^* if and only if the sets $S_{i,\alpha}(y) := \{x | a_i^t x = b_i\} \cap \{x | F(x, y) \leq \alpha\}$ are nonempty for all $y \in M(\alpha)$.*

Proof: " \Rightarrow ". Obviously $x^* \in S_{i,\alpha}(y)$ for all $y \in M(\alpha)$. " \Leftarrow ". For $y \in M(\alpha)$, let $z_i^*(y)$, be a point satisfying $a_i^t z_i^*(y) = b_i$ and $F(y, z_i^*(y)) \leq \alpha$. Assuming that the i -th constraint is non-active at x^* , then, by Prop. 2, there is $y_0 \in M(\alpha)$ s.t. $i \in J_-(y_0)$ and $r_i^2(y_0) > 2\alpha q_{ii}$. But $r_i^2(y_0) = 2F(y_0, z_i^*(y_0))q_{ii}$, i.e. $F(y_0, z_i^*(y_0)) > \alpha$, a contradiction. ■

3 Sufficient Conditions for Superfluous Constraints

At an arbitrary point $y \in M(\alpha)$ all restrictions i , satisfying $\beta_i(y) := F(y, z_i^*(y)) = \frac{r_i^2(y)}{2q_{ii}} > \alpha$ are identified to be superfluous for **QP**. To get additional non-active constraints within the remaining ones consider two nonlinear optimization problems at y : $\min_{x \in S_{i,\alpha}(y)} a_j^t x$ and $\max_{x \in S_{i,\alpha}(y)} a_j^t x$. Let their optimal values be denoted by $\tau_{j,i}^*(\alpha, y)$ and $\tau_{j,i}^{**}(\alpha, y)$, respectively.

Proposition 4.

- i. $\tau_{j,i}^*(\alpha, y) = a_j^t y - q_{ii}^{-1} \left[r_i(y)q_{ij} + (2\alpha q_{ii} - r_i(y)^2)^{\frac{1}{2}}(q_{ii}q_{jj} - q_{ij}^2)^{\frac{1}{2}} \right]$.
- ii. $\tau_{j,i}^{**}(\alpha, y) = a_j^t y - q_{ii}^{-1} \left[r_i(y)q_{ij} - (2\alpha q_{ii} - r_i^2(y))^{\frac{1}{2}}(q_{ii}q_{jj} - q_{ij}^2)^{\frac{1}{2}} \right]$.
- iii. If $\tau_{j,i}^*(\alpha, y) > b_j$, then the i -th restriction is superfluous for **QP**.
- iv. If $\tau_{j,i}^{**}(\alpha, y) < b_j$, then either the i -th or the j -th restriction is superfluous for **QP**.

Proof: (i.) Note, that both optimization problems have solutions since **QP** is assumed to be solvable and $\beta_i(y) \leq \alpha$. The proofs of *i.* and *ii.* are then based on the same construction used in [3], up to exchanging x_0 and y on the one hand and $f(x)$ and $F(y, x)$ on the other.

(ii.) The condition $\tau_{j,i}^*(\alpha, y) > b_j$ implies, that $a_j^t \bar{x} \geq \tau_{j,i}^*(\alpha, y) > b_j$ for every point $\bar{x} \in S_{i,\alpha}(y)$. Hence, $x^* \notin S_{i,\alpha}(y)$. But $x^* \in \{x | F(x, y) \leq \alpha\}$. Therefore, we conclude that $a_i^t x^* < b_i$.

iv.) The assumption $a_i^t x^* = b_i$ would imply $a_j x^* \leq \tau_{j,i}^{**}(\alpha, y) < b_j$. I.e., both restrictions cannot be active at x^* simultaneously. ■

If a_i and a_j are linearly independent (i.e. $q_{ii} \cdot q_{jj} > q_{ij}^2$), the function $\tau_{j,i}^*(\alpha, y)$ is strictly decreasing on $[\beta_i(y), \infty[$, with $\tau_{j,i}^*(\alpha, y) \rightarrow -\infty$ if $\alpha \rightarrow \infty$. The function $\tau_{j,i}^{**}(\alpha, y)$ is strictly increasing on $[\beta_i(y), \infty[$ and $\tau_{j,i}^{**}(\alpha, y) \rightarrow +\infty$ if $\alpha \rightarrow \infty$.

In this case we get

Corollary 2. For $y \in M(\alpha)$ let $\alpha_{i,j}^*(y) := (2q_{ii})^{-1}[r_i^2(y) + (r_j(y)q_{ii} - r_i(y)q_{ij})^2 \cdot (q_{ii}q_{jj} - q_{ij}^2)^{-1}]$. Assume that $\alpha < \alpha_{i,j}^*(y)$.

i. If $r_j(y)q_{ii} - r_i(y)q_{ij} \geq 0$, then the *i* – th restriction is non-active at x^* .

ii. If $r_j(y)q_{ii} - r_i(y)q_{ij} < 0$, then either the *i* – th or the *j* – th restriction is non-active at x^*

Proof: An easy calculation shows, that the equation $\tau_{j,i}^*(\alpha, y) = b_j$ is solved by $\alpha_{i,j}^*(y)$ if and only if $r_j(y)q_{ii} - r_i(y)q_{ij} \geq 0$. Similar, $\tau_{j,i}^{**}(\alpha, y) = b_j$ has the solution $\alpha_{i,j}^*(y)$ if and only if $r_j(y)q_{ii} - r_i(y)q_{ij} \leq 0$. Now, with the last proposition we get (i.) and (ii.), respectively. ■

Since condition *i.* and *ii.* of corollary 2 exclude each other they induce a partition of $M(\alpha) = M_+(\alpha) \cup M_-(\alpha)$, where $M_+(\alpha) := \{y \in M(\alpha) | r_j(y)q_{ii} - r_i(y)q_{ij} \geq 0\}$ and $M_-(\alpha) := \{y \in M(\alpha) | r_j(y)q_{ii} - r_i(y)q_{ij} < 0\}$. Hence, choosing some $y \in M(\alpha)$ and checking the condition $\alpha < \alpha_{i,j}^*(y)$, we can either eliminate all those restrictions satisfying *i.* or, if *ii.* is satisfied, at least one of the constraints *i* and *j* will be superfluous at x^* . So, not both can appear in an active set at x^* . We will put them to some *watch list* for a possible later decision which of them can be omitted.

In the opposite case a_i and a_j are linearly dependent (i.e. $q_{ii} \cdot q_{jj} = q_{ij}^2$) and every point in $S_{i,\alpha}(y)$ is an optimal solution with optimal value $\tau_{j,i}^*(\alpha, y) = \tau_{j,i}^{**}(\alpha, y) = a_j^t y - q_{ii}^{-1} r_i(y)q_{ij}$.

In this case we have

Corollary 3. If $\tau_{j,i}^{**}(y) = b_j$, then at least one of the restrictions *i* or *j* is not necessary for the determination of x^* . If $q_{ij} < 0$, then both restrictions are active at x^* .

Proof: The intersection of the two affine half-spaces $\{x | a_i^t x \leq b_i\}$ and $\{x | a_j^t x \leq b_j\}$ either describe $\{x | a_i^t x \leq \min\{b_i, \frac{\|a_i\|}{\|a_j\|} b_j\}\}$, or $\{x | -\frac{\|a_i\|}{\|a_j\|} b_j \leq a_i^t x \leq b_i\}$. The first case occurs if $q_{ij} > 0$. In this case one of the half-spaces is contained in the other, so that least one of the corresponding constraints

is superfluous for QP . In the second case $q_{ij} < 0$. For every $x \in S_{i,\alpha}(y)$ we get $b_j = \tau_{j,i}^*(\alpha, y) \leq a_j^t x \leq \tau_{j,i}^{**}(\alpha, y) = b_j$, i.e. $x \in S_{j,\alpha}(y)$. This implies $b_i = -\frac{\|a_i\|}{\|a_j\|} b_j$, i.e. both constraints must be active at x^* . ■

If we use a QP -solver that produces a decreasing sequence of upper bounds α , all the conditions suggest to look for "large values" of $\beta_i(y), \alpha_{ij}^*(y)$. As larger these values are, as earlier superfluous constraints can be detected. Largest values for these quantities can be found on the boundary of $M(\alpha)$, if a_i, a_j are linearly independent:

Lemma 1. (i.) *The point $z_i^* := \operatorname{argmax}_{y \in M(\alpha)} \beta_i(y)$ is a boundary point of $M(\alpha)$.*

(ii.) *If $M_+(\alpha) \neq \emptyset$, then $y^* := \operatorname{argmax}_{y \in M_+(\alpha)} \alpha_{ij}^*(y)$ is a boundary point of $M(\alpha)$. If $M_-(\alpha) \neq \emptyset$, then $y^{**} := \operatorname{argmax}_{y \in M_-(\alpha)} \alpha_{ij}^*(y)$ has this property.*

Proof: (i.) Follows by simple convexity arguments.

(ii.) Consider $y^* := \operatorname{argmax}_{y \in M_+(\alpha)} \alpha_{ij}^*(y)$. We distinguish two cases.

First, we assume that $r_j(y^*) = q_{ii}^{-1} q_{ij} r_i(y^*)$. In this case $\alpha_{i,j}^*(y^*) = \beta_i(y^*) = \alpha > 0$, as in (i.).

In the second case, i.e. $r_j(y^*) < q_{ii}^{-1} q_{ij} r_i(y^*)$, there is a neighborhood $U(y^*)$ of y^* , s.t. all $y \in U(y^*)$ solve $\tau_{j,i}^*(\alpha, y) = b_j$.

Now, assume, that $F(x^*, y^*) < \alpha$. Since $\alpha_{ij}^*(y)$ is continuously differentiable on $U(y^*)$, we get $\nabla \alpha_{ij}^*(y^*) = 0$, i.e. $0 = a_i \left[\frac{r_i(y^*)}{q_{ii}} - \frac{q_{ij}(r_j(y^*)q_{ii} - r_i(y^*)q_{ij})}{q_{ii}(q_{ii}q_{jj} - q_{ij}^2)} \right] + a_j \left[\frac{r_j(y^*)q_{ii} - r_i(y^*)q_{ij}}{q_{ii}q_{jj} - q_{ij}^2} \right]$. We conclude $r_j(y^*) = r_i(y^*) = 0$, contradicting $r_j(y^*) < q_{ii}^{-1} q_{ij} r_i(y^*)$. ■

4 Constructing Points in $M(\alpha)$.

Although propositions 2 and 3 completely characterize the set of active and non-active constraints, the set $M(\alpha)$ is not explicitly given, since x^* is unknown in advance. Nevertheless, there is an algorithmic scheme that allows to construct at least subsets of $M(\alpha)$.

Proposition 5. *Let $y \in M(\alpha)$ and $k \in J_+(y)$. For $\gamma \geq 0$ let $y(\gamma) := y - \gamma \cdot \frac{r_k(y)}{q_{kk}} \cdot C^{-1} a_k$.*

(i.) *If $\gamma \in [0, 2]$, then $y(\gamma) \in M(\alpha)$.*

(ii.) *If $\gamma \in]1, 2]$, then $k \in J_-(y(\gamma))$.*

Proof: (i.) $F(x^*, y(\gamma)) = \frac{1}{2}(x^* - y(\gamma))^t C(x^* - y(\gamma))$
 $= \frac{1}{2}(x^* - y)^t C(x^* - y) + \gamma \cdot \frac{r_k(y)}{q_{ii}} (x^* - y)^t a_k + \frac{1}{2} \gamma^2 \cdot \left[\frac{r_k(y)^2}{q_{kk}} \right]$
 $= \frac{1}{2}(x^* - y)^t C(x^* - y) + \gamma \cdot \frac{r_k(y)}{q_{kk}} [(x^* - y)^t a_k + \frac{1}{2} \gamma \cdot r_k(y)].$

But now, $0 \leq \gamma r_k(y) = \gamma(a_k^t y - b_k) \leq \gamma(a_k^t y - a_k^t x^*) \leq 2(a_k^t y - a_k^t x^*)$.

Hence $(x^* - y)^t a_k + \frac{1}{2} \gamma \cdot r_k(y) \leq 0$. This implies $F(x^*, y(\gamma)) \leq \alpha$.

(ii.) follows, since $r_k(y(\gamma)) = (1 - \gamma)r_k(y)$. ■

Since $\alpha \geq f(x^*)$ and $x_0 \in M(f(x^*))$, the proposition suggests a procedure to exhaust the unknown set $M(\alpha)$. It will produce a set of points, say $A(x_0)$. Each $y \in A(x_0)$ then might serve as a "checkpoint" to detect superfluous restrictions for QP.

Procedure A(x_0):

Step 0: Set $y_0 := x_0$ and $j := 0$

Step 1: Choose $k \in \{C^{-1}a_k | 1 \leq k \leq m, k \in J_+(y_j)\}$ arbitrarily.

Choose $\gamma_j \in [0, 2]$ arbitrarily.

Compute new iterate

$$y_{j+1} := y_j - \gamma_j \cdot \frac{r_k(y_j)}{q_{kk}} \cdot C^{-1}a_k.$$

Step 2: If $y_{j+1} \in S$, Stop.

Otherwise, set $j := j + 1$ and go to Step 1

Different points y_{j+1} can be computed by choosing different $k \in J_+(y_j)$ or by the different choice of γ_j . The construction of points y_{j+1} obviously does not depend on the upper bound α , i.e. all that points are all elements of the (unknown) set $M(f(x^*))$. Instead of looking for best possible points z^*, y^*, y^{**} on the boundary of $M(f(x^*))$, the values of $\beta_i(y)$ and $\alpha_{i,j}^*(y)$ can successively be improved within $A(x_0)$. Such an improvement might be done by moving from y to $y(2)$. For the sake of abbreviation, let us define additional structural quantities $c_{ijk} := q_{jk}q_{ii} - q_{ik}q_{ij}$ and $b_{ijk} := c_{ijk}(q_{ii}q_{jj} - q_{ij}^2)$

Proposition 6. Let $y \in A(x_0)$, $i \in \{1, 2, \dots, m\}$ and $y(\gamma) := y - \gamma \cdot \frac{r_k(y)}{q_{kk}} \cdot C^{-1}a_k$ for some $k \in J_+(y)$ and $\gamma \geq 0$. Let $q_{ik} \neq 0$.

(i.) There is $\gamma \in [0, 2]$ such that $\beta_i(y(\gamma)) > \beta_i(y)$ if and only if $\frac{r_i(y)q_{kk}}{r_k(y)q_{ik}} < 1$.

In this case $\beta_i(y(\gamma))$ is maximized at $\gamma = 2$.

(ii.) There is $\gamma \in [0, 2]$ such that $\alpha_{i,j}^*(y(\gamma)) > \alpha_{i,j}^*(y)$, if and only if

$$\frac{q_{kk}}{r_k(y)} \left(\frac{r_j(y)c_{ijk} + r_i(y)c_{jik}}{q_{jk}c_{ijk} + q_{ik}c_{jik}} \right) < 1. \text{ In this case } \alpha_{i,j}^*(y(\gamma)) \text{ is maximized at } \gamma = 2.$$

Proof: (i.) Obviously, $r_i^2(y(\gamma)) = r_i^2(y) - 2\gamma r_i(y)r_k(y)\frac{q_{ik}}{q_{kk}} + \gamma^2 r_k^2(y)\left(\frac{q_{ik}}{q_{kk}}\right)^2$.

Hence for some $\gamma \in [0, 2]$ we have, $\beta_i(y(\gamma)) > \beta_i(y)$ if and only if $2 \geq \gamma > 2\frac{r_i(y)q_{kk}}{r_k(y)q_{ik}}$. Since $r_i^2(y(\gamma))$ is strictly increasing for $\gamma > \frac{r_i(y)q_{kk}}{r_k(y)q_{ik}}$ the best possible value is attained at $\gamma_0 = 2$.

(ii.) For $\lambda \in R$ consider the function $\alpha_{i,j}^*(y(\lambda)) := (2q_{ii})^{-1}[r_i^2(y(\lambda)) + (r_j(y(\lambda))q_{ii} -$

$r_i(y(\lambda))q_{ij})^2 \cdot (q_{ii}q_{jj} - q_{ij}^2)^{-1}]$, where $y(\lambda) = y - \lambda \cdot \frac{r_k(y)}{q_{kk}} \cdot C^{-1}a_k$. It is a convex

function and attains its (global) minimal point at $\lambda^* = \frac{q_{kk}}{r_k(y)} \left(\frac{r_j(y)c_{ijk} + r_i(y)c_{jik}}{q_{jk}c_{ijk} + q_{ik}c_{jik}} \right)$.

By monotonicity of the function $\alpha_{i,j}^*(y(\lambda))$, we get $\alpha_{i,j}^*(y(\lambda)) < \alpha_{i,j}^*(y(2))$ for all $\lambda \in [0, 2]$, if and only if $\lambda^* < 1$.

In the opposite case, i.e. $\lambda^* \geq 1$, then $\alpha_{i,j}^*(y(\lambda)) \leq \alpha_{i,j}^*(y(0))$ for all $\lambda \in [0, 2]$. ■

Points $y \in A(x_0)$ might also provide lower bounds on the optimal value of QP.

Proposition 7. Let $y \in M(f(x^*)), i \in J_+(y)$.

(i.) $\beta_i(y) \leq F(x^*, y) \leq f(x^*)$.

(ii.) If there is $\bar{\alpha}$ such that $\tau_{j,i}^*(\bar{\alpha}, y) > b_j$ for some $j \notin J_+(y)$, then $\bar{\alpha} \leq f(x^*)$.

Proof: i. $f(x^*) = F(x_0, x^*) \geq F(y, x^*) \geq \min_{z \in \{x | a_i^t x = a_i^t x^*\}} F(y, z) \geq \min_{z \in \{x | a_i^t x = b_i\}} F(y, z) = \beta_i(y)$.

ii. Obviously, there is $\lambda \in [0, 1[$, $x_i = \lambda y + (1 - \lambda)x^* \in M(f(x^*))$ with $a_i^t x_i = b_i$. Since $j \notin J_+(y)$, $a_j^t x_i \leq b_j$. The assumption $f(x^*) < \bar{\alpha}$ yields $F(y, x_i) \leq F(y, x^*) \leq F(x_0, x^*) = f(x^*) < \bar{\alpha}$, i.e. $x_i \in S_{i, \bar{\alpha}}(y)$. But this implies $\tau_{j,i}^*(\bar{\alpha}, y) \leq b_j$, a contradiction. ■

Last, but not least we will show, that the optimal point x^* for **QP** is a limit point of $A(x_0)$. We first prove

Lemma 2. Let x^* be the optimizer of $\min_{x \in S} F(x, y)$ and denote by I_{act} the set of active constraints at x^* . If $y \notin S$ then $I_{act} \cap J_+(y) \neq \emptyset$.

Proof: Assume the contrary, i.e. $a_j^t y \leq b_j$ for all $j \in I_{act}$. Consider points $\bar{x}(\sigma) := \sigma y + (1 - \sigma)x^*, \sigma \in [0, 1]$. Choose $i \in \{1, 2, \dots, m\}$. If $i \in I_{act}$, $a_i^t \bar{x}(\sigma) \leq b_i$, for all $\sigma \in [0, 1]$. The same is true, if $i \notin I_{act}$, but $a_i^t y \leq b_i$.

The last case that can happen is, that i satisfies $a_i^t y > b_i$ but $a_i^t x^* < b_i$. Denote by I the set of those constraints. For each $i \in I$ there is $\sigma_i \in]0, 1[$ with $a_i^t \bar{x}(\sigma) < b_i$ for all $\sigma \in [\sigma_i, 1]$. Let $\sigma^* := \max_{i \in I} \sigma_i$. Obviously, $\bar{x}(\sigma^*) \in S$. But $0 < \sigma^* < 1$, implying that $F(\bar{x}(\sigma^*), y) < F(x^*, y)$, contradicting the optimality of x^* . ■

Proposition 8. x^* is a limit point of $A(x_0)$

Proof: First note, that if the optimal point x^* of **QP** is also optimal for the problem $\min_{x \in S} F(x, y)$, then there are multipliers $\lambda_i = \lambda_i(y) \geq 0$ such that $\nabla_x F(x, y)|_{x^*} = -\sum_{i \in I_{act}} \lambda_i a_i$, and $\sum_{i \in I_{act}} \lambda_i (a_i^t x^* - b_i) = 0$. I.e. $x^* = y - \sum_{i \in I_{act}} \lambda_i C^{-1} a_i$. If $y \notin S$, lemma 2 gives $I_{act} \cap J_+(y) \neq \emptyset$. Moreover, there is at least one constraint, say $k \in I_{act} \cap J_+(y)$, such that $\lambda_k > 0$. This must be true, since otherwise $0 < (x^* - y)^t C(x^* - y) = -\sum_{i \in I_{act}} \lambda_i a_i^t (x^* - y) = -\sum_{i \in I_{act} \cap J_-} \lambda_i (b_i - a_i^t y) \leq 0$, would produce a contradiction.

Now, consider an arbitrary point $y_j \in M(\alpha) \setminus S$ such that x^* minimizes $F(y_j, x)$ on S . By lemma 1, there is $k \in J_+(y_j) \cap I_{act}$ with $\lambda_k > 0$. Set $c_j := \min\{\frac{\lambda_k(y_j) q_{kk}}{r_k(y_j)}, 2\}$ and chose $\gamma_j \in [0, c_j]$ arbitrarily. Define $y_{j+1} := y_j - \gamma_j \frac{r_k(y_j)}{q_{kk}} C^{-1} a_k$, $\lambda_k(y_{j+1}) := \lambda_k(y_j) - \gamma_j \frac{r_k(y_j)}{q_{kk}}$, $\lambda_i(y_{j+1}) := \lambda_i$ for $i \neq k$. The point y_{j+1} together with the multipliers $\lambda_i(y_{j+1}) \geq 0$ then satisfy $C(x^* - y_{j+1}) = -\sum_{i \in \{1, 2, \dots, m\}} \lambda_i(y_{j+1}) a_i$, and $\sum_{i \in \{1, 2, \dots, m\}} \lambda_i(y_{j+1}) (a_i^t x^* - b_i) = 0$. I.e. x^* is also the optimizer of the problem $\min_{x \in S} F(x, y_{j+1})$. Obviously, $y_{j+1} \in A(y_j)$. If $y_{j+1} \in S$, then $F(x, y_{j+1})$ is minimized on S by y_{j+1} , i.e. $y_{j+1} = x^*$. In the opposite case $y_{j+1} \notin S$ and the above construction can be repeated using y_{j+1} instead of y_j .

Now, starting with $y_1 := x_0$ this procedure will either terminate after a finite number N of steps at a feasible point $y_N = x^* \in A(x_0)$, or it will produce

an infinite sequence $(y_j)_{j=1,2,\dots} \subset M(\alpha)$, $y_j \in \mathbf{A}(x_0)$ of non-feasible points. Let D be the set of all limit points that can be obtained by the above construction method, i.e the set of points \bar{y} , for which there is a sequence $(y_j)_{j=1,2,\dots}$, constructed in the above way and a subset $\mathcal{A} \subset \{1, 2, \dots\}$ such that $\lim_{j \in \mathcal{A}} y_j = \bar{y}$.

Note, that the related multipliers $\lambda_i(y_j)$, $j \in \mathcal{A}$ then induce limit points $\bar{\lambda}_i$. For fixed i the sequence $(\lambda_i(y_j))_{j=1,2,\dots}$ is monotonously decreasing with greatest lower bound $\bar{\lambda}_i \geq 0$. Using the same arguments as above, x^* is the optimizer for $\min_{x \in S} F(x, \bar{y})$. For the case $\bar{y} \in S$ this again leads to $x^* = \bar{y} \in D \subset cl(\mathbf{A})$. Hence, it remains to show, that $D \cap S \neq \emptyset$.

For this, suppose $D \cap S = \emptyset$. Since D is closed and $D \subset M(\alpha)$, the function $F(x^*, y)$ attains its minimum at some $\bar{y} \in D$. I.e. $0 < F(x^*, \bar{y}) \leq F(x^*, y)$ for all $y \in D$. Since $\bar{y} \notin S$, by lemma 2 there is $k \in J_+(\bar{y}) \cap I_{act}$ with $\bar{\lambda}_k > 0$. Moreover, there is j_0 such that $r_k(y_j) > \frac{r_k(\bar{y})}{2} > 0$ for all $j \geq j_0, j \in \mathcal{A}$. On the other hand $r_k(y_j) \leq \bar{R}$ for some $\bar{R} > 0$.

Now, let $\epsilon > 0$ be chosen in such a way, that $0 < \epsilon < \frac{1}{2} q_{kk} r_k^2(\bar{y}) \cdot \min\{2, \frac{\bar{\lambda}_k q_{kk}}{\bar{R}}\}$. Then there is j_1 such that for all $j \geq j_1, j \in \mathcal{A}$ the inequality $F(x^*, y_j) < F(x^*, \bar{y}) + \frac{\epsilon}{8}$ holds. Now, chose $\gamma > 0$, satisfying $\min\{2, \frac{\bar{\lambda}_k q_{kk}}{\bar{R}}\} \geq \gamma > \frac{2\epsilon q_{kk}}{r_k^2(\bar{y})}$, fix $j \geq \max\{j_0, j_1\}$, $j \in \mathcal{A}$ and set $\tilde{y} := y_j - \gamma \cdot \frac{r_k(y)}{q_{kk}} \cdot C^{-1} a_k$.

Obviously, $\gamma \leq \frac{\bar{\lambda}_k q_{kk}}{\bar{R}} \leq \frac{\lambda_k(y_j) q_{kk}}{r_k(y_j)}$, i.e. $\tilde{y} \in D$. With $F(x^*, \tilde{y}) = F(x^*, y_j) - \epsilon \frac{r_k^2(y_j)}{r_k^2(\bar{y})} < F(x^*, y_j) - \frac{\epsilon}{4} < F(x^*, \bar{y}) - \frac{\epsilon}{4} + \frac{\epsilon}{8} = F(x^*, \bar{y}) - \frac{\epsilon}{8}$, we then get the contradiction, that \bar{y} is a minimizer of $F(x^*, x)$ on D . ■

References

1. Goldfarb, D., Liu, S. (1991) An $O(n^3 L)$ primal interior point algorithm for convex quadratic programming. Math. Prog. Ser. A, 3, 325-340
2. Mehrotra, S., Sun, J. (1990) An algorithm for convex quadratic programming that requires $O(n^{3.5} L)$ arithmetic operations. Math. Oper. Res. 15, No. 2, 342-363.
3. Recht, P. (2001) Identifying non-active restrictions in convex quadratic programming. Math. Meth. of OR 54, 53-61.
4. Recht, P. (2003) Redundancies in positive semidefinite quadratic programming. JOTA. 119, No. 3, 553-564.

Fast Optimal Control Algorithms with Application to Chemical Engineering

Andreas Schäfer*, Ulrich Brandt-Pollmann, Moritz Diehl,
Hans-Georg Bock, and Johannes P. Schlöder

Interdisciplinary Center for Scientific Computing (IWR), University of
Heidelberg, Germany

Abstract. A comparison of two direct multiple shooting approaches for the solution of chemical optimization problems with DAE models of index one is presented. Differences are explained in terms of the underlying reduced SQP methods. The application to optimal control of a distillation column is used to illustrate the respective advantages.

1 Introduction

Optimal design and operation of complex chemical processes are challenging tasks for mathematical optimization tools. Direct simultaneous approaches have proven to be good for optimization problems with large, sparse DAE process models of index one. Especially discretizing the dynamic model by the multiple shooting technique allows the use of existing advanced, fully adaptive DAE solvers. In order to cope with inconsistent algebraic start values relaxation techniques can be used which allow the use of infeasible SQP approaches for all state variables. In the following two direct multiple shooting approaches for the solution of optimal control problems with large, sparse DAE process models are presented. They differ in the type of reduction technique in the SQP method. The first one projects the structured NLP problem onto the reduced space of differential variables plus control parameters, using the fact that the algebraic variables are uniquely determined by the consistency conditions due to the index one assumption. The second one additionally uses initial conditions and continuity conditions for projection onto the reduced space of control parameters. In both cases it is demonstrated that the expensive calculation of the full set of differential state derivatives can be avoided. Instead it suffices to compute only a reduced number of directional derivatives. The approaches are compared in terms of the number of directional derivatives. In order to calculate the sparse model Jacobians an algorithmic differentiation tool is used in combination with seed matrix compression. The sparsity of the model Jacobians is exploited in the DAE solver and directional differential state derivatives are calculated using the

* e-mail: Andreas.Schaefer@IWR.Uni-Heidelberg.De, Brandt-Pollmann@IWR,
M.Diehl@IWR, Bock@IWR, J.Schloeder@IWR

principle of Internal Numerical Differentiation (IND). As an application we present the optimal control of a distillation column for separation of a binary mixture. Results are presented which compare the computing times for both approaches.

2 Problem Formulation

Many dynamic process optimization problems in chemical engineering can be expressed as multistage optimal control problems in DAE. Throughout this paper, we will consider the following general class of M -stage optimal control problems, where the time horizon of interest $[t_0, t_M]$ is divided into M model stages corresponding to the subintervals $[t_i, t_{i+1}]$, $i = 0, 1, \dots, M-1$.

In the following x_i are the differential states, z_i are the algebraic states, u_i are the controls, p are the global parameters and t_i are time points of the time horizon, with i signifying the respective model stage.

The performance index may be of Bolza type

$$\min_{x_i, z_i, u_i, p, t_i} \sum_{i=0}^{M-1} \left(\int_{t_i}^{t_{i+1}} \Phi_i(x_i(t), z_i(t), u_i(t), p, t) dt + \phi_i(x_i(t_{i+1}), z_i(t_{i+1}), p, t_{i+1}) \right) \quad (1a)$$

subject to the DAE model stages

$$\left. \begin{aligned} B_i(\cdot) \dot{x}_i(t) &= f_i(x_i(t), z_i(t), u_i(t), p, t) \\ 0 &= g_i(x_i(t), z_i(t), u_i(t), p, t) \end{aligned} \right\}, \quad t \in [t_i, t_{i+1}], \quad i = 0, 1, \dots, M-1, \quad (1b)$$

the control and path constraints

$$h_i(x_i(t), z_i(t), u_i(t), p, t) \geq 0, \quad t \in [t_i, t_{i+1}], \quad i = 0, 1, \dots, M-1. \quad (1c)$$

the stage transition conditions

$$x_{i+1}(t_{i+1}) = c_i(x_i(t_{i+1}), z_i(t_{i+1}), p, t_{i+1}), \quad i = 0, 1, \dots, M-2. \quad (1d)$$

and the multipoint boundary conditions

$$\sum_{i=0}^{M-1} (r_i^s(x_i(t_i), z_i(t_i), p, t_i) + r_i^e(x_i(t_{i+1}), z_i(t_{i+1}), p, t_{i+1})) \left\{ \begin{aligned} &= \\ &\geq \end{aligned} \right\} 0. \quad (1e)$$

3 Two Direct Multiple Shooting Methods for DAEs

The original continuous problem is reformulated as a NLP problem with a finite number of optimization variables. On each model stage $i = 0, 1, \dots, M-1$,

we employ the time transformation $\theta_i(\tau, v) := t_i + \tau d_i$, $t_i = t_0 + \sum_{k=0}^{i-1} d_k$, $\tau \in [0, 1]$ with $v := (t_0, d_0, d_1, \dots, d_{M-1})$, and we choose a fixed, dimensionless discretization grid

$$0 = \tau_{i,0} < \tau_{i,1} < \dots < \tau_{i,m_i} = 1. \quad (2)$$

Then we define a piecewise approximation \hat{u}_i of the control vector u_i by

$$\hat{u}_i(\tau) := \varphi_{ij}(\tau, q_{ij}), \quad \tau \in I_{ij} = [\tau_{ij}, \tau_{i,j+1}], \quad j = 0, 1, \dots, m_i - 1, \quad (3)$$

using “local” control parameters q_{ij} . The functions φ_{ij} are given basis functions, typically vectors of polynomials (e.g., constant, linear, or cubic).

The DAEs (1b) are discretized on the same grid (2) by a specific variant of *multiple shooting* which has been proposed by Bock *et al.* (1988 [6]). We introduce additional optimization parameters s_{ij}^x , s_{ij}^z and solve the following set of relaxed decoupled initial value problems (IVPs)

$$\left. \begin{aligned} B_i(\cdot) dx_i(\tau)/d\tau &= f_i(x_i(\tau), z_i(\tau), \varphi_{ij}(\tau, q_{ij}), p, \theta_i(\tau, v)) d_i \\ 0 &= g_i(x_i(\tau), z_i(\tau), \varphi_{ij}(\tau, q_{ij}), p, \theta_i(\tau, v)) \\ &\quad - \alpha_{ij}(\tau) g_i(s_{ij}^x, s_{ij}^z, \varphi_{ij}(\tau_{ij}, q_{ij}), p, \theta_i(\tau_{ij}, v)) \end{aligned} \right\}, \quad \tau \in I_{ij}, \quad (4)$$

using $x_i(\tau_{ij}) = s_{ij}^x$ and $z_i(\tau_{ij}) = s_{ij}^z$ as initial values. The scalar damping factor α_{ij} is given by a scalar function which is non-increasing and non-negative on I_{ij} and satisfies $\alpha_{ij}(\tau_{ij}) = 1$. Let $x_i(\tau_{i,j+1}; s_{ij}^x, s_{ij}^z, q_{ij}, p, v)$ denote the differential state values at $\tau = \tau_{i,j+1}$ obtained by numerical integration of (4). The integral terms of the performance index are reformulated as a Mayer objective with the introduction of an additional differential state. The control and path constraints are discretized on the multiple shooting grid (2). Now the complete reformulation of the continuous M -stage optimal control problem leads to the NLP problem

$$\min_{s_{ij}^x, s_{ij}^z, q_{ij}, p, v} \sum_{i=0}^{M-1} \phi_i(s_{i,m_i}^x, s_{i,m_i}^z, p, \theta_i(\tau_{i,m_i}, v)) \quad (5a)$$

subject to the continuity and consistency conditions

$$\left. \begin{aligned} x_i(\tau_{i,j+1}; s_{ij}^x, s_{ij}^z, q_{ij}, p, v) - s_{i,j+1}^x &= 0, \quad j = 0, 1, \dots, m_i - 1 \\ g_i(s_{ij}^x, s_{ij}^z, \varphi_{ij}(\tau_{ij}, q_{ij}), p, \theta_i(\tau_{ij}, v)) &= 0, \quad j = 0, 1, \dots, m_i \end{aligned} \right\}, \quad \dots, M-1, \quad (5b)$$

the discretized control and path constraints

$$h_i(s_{ij}^x, s_{ij}^z, \varphi_{ij}(\tau_{ij}, q_{ij}), p, \theta_i(\tau_{ij}, v)) \geq 0, \quad \left\{ \begin{aligned} j &= 0, 1, \dots, m_i, \\ i &= 0, 1, \dots, M-1, \end{aligned} \right. \quad (5c)$$

the stage transition conditions

$$c_i(s_{i,m_i}^x, s_{i,m_i}^z, p, \theta_i(\tau_{i,m_i}, v)) - s_{i+1,0}^x = 0, \quad i = 0, 1, \dots, M-2, \quad (5d)$$

and the linearly coupled multipoint constraints

$$\sum_{i=0}^{M-1} (r_i^s(s_{i,0}^x, s_{i,0}^z, p, \theta_i(\tau_{i,0}, v)) + r_i^e(s_{i,m_i}^x, s_{i,m_i}^z, p, \theta_i(\tau_{i,m_i}, v))) \left\{ \begin{matrix} = \\ \geq \end{matrix} \right\} 0. \quad (5e)$$

The global model parameters (p, v) are “localized” by introducing new NLP variables $(p, v)_{ij}$ together with additional linear constraints enforcing $(p, v)_{ij} - (p, v)_{0,0} = 0$ for all $(i, j) \neq (0, 0)$. Then only *linear coupling* occurs between variables corresponding to different discretization intervals.

This structured NLP problem (5) is solved by specially tailored SQP (Bock, Plitt 1984 [4], Leineweber 1999 [11]) and Gauss-Newton methods (Bock 1981 [2], 1983 [3], 1987 [5], Schlöder 1988 [13]) in case of general or Least-Squares functions in the objective respectively. In order to motivate the later presented algorithms a general concept of a reduced SQP (see Gabay 1982 [9], Schulz 1996 [14], Leineweber 1999 [11]) method is introduced for the following NLP problem

$$\min_w F(w) \quad \text{s. t.} \quad \begin{cases} G_1(w) = 0 \\ G_2(w) = 0 \\ H(w) \geq 0 \end{cases}. \quad (6)$$

This problem is iteratively solved by computing the SQP iterates

$$w_{k+1} = w_k + \alpha_k \Delta w_k \quad (7)$$

where α_k is used for globalization by e.g. line search. The calculation of the step Δw_k is divided into two parts through introduction of a coordinate basis (S_k^N, S_k^R)

$$\Delta w_k = \underbrace{\begin{pmatrix} -\nabla_{w_1} G_{1,k}^{T-1} \nabla_{w_2} G_{1,k}^T \\ I \end{pmatrix}}_{:=S_k^N} y_k^N + \underbrace{\begin{pmatrix} I \\ 0 \end{pmatrix}}_{:=S_k^R} y_k^R, \quad (8)$$

where we use a partitioning of the NLP variables $w \equiv (w_1, w_2)$ such that $\nabla_{w_1} G_{1,k}^{T-1}$ is nonsingular. Note that the linearized constraints of G_1 are used to set up this coordinate basis.

The range-space component y_k^R is defined by

$$y_k^R := -\nabla_{w_1} G_{1,k}^{T-1} G_{1,k} \quad (9)$$

and the null-space component y_k^N is the solution of the following reduced QP:

$$\min_{y_k^N} \nabla F_k^T S_k^N y_k^N + \underbrace{y_k^{RT} S_k^R \nabla_{w_k}^2 L_k S_k^N y_k^N}_{\text{crossterm}} + \frac{1}{2} y_k^{NT} \underbrace{S_k^{NT} \nabla_{w_k}^2 L_k S_k^N}_{\text{red. Hessian}} y_k^N \quad (10)$$

$$\text{s. t.} \quad \begin{cases} G_{2,k} + \nabla G_{2,k}^T S_k^R y_k^R + \nabla G_{2,k}^T S_k^N y_k^N = 0 \\ H_k + \nabla H_k^T S_k^R y_k^R + \nabla H_k^T S_k^N y_k^N \geq 0 \end{cases}.$$

Here L_k is the Lagrange function for the NLP problem (6). Two remarks are in order: first, a good approximation of the crossterm is easily available if the objective function is of Least-Squares type. In this case Gauss-Newton Hessians are used which give a direct mathematical equivalence of a reduced space algorithm and a full-space algorithm. Second the reduced Hessian for general objective function will be usually approximated by BFGS-Updates for which the difference of the reduced Lagrange gradients $S_{k+1}^N T \nabla L_{k+1} - S_k^N T \nabla L_k$ and the null-space step y_k^N are needed to define the update between w_k and w_{k+1} .

In the following two different levels of reduced space algorithms are presented. The first one is characterized by defining $G_{1,k}$ as the consistency conditions, $G_{2,k}$ as the continuity, stage transition and multipoint equality conditions and H_k as the discretized control and path constraints and multipoint inequality conditions. This has several consequences: as in the full-space variant the Hessian of the reduced KKT system is block diagonal. Also the integrations on different multiple shooting intervals are completely decoupled and can therefore be performed in parallel. This algorithm was developed by Leineweber 1999 [11], and implemented in the code MUSCOD-II. This reduced space variant has high computation times for process models described by PDEs and discretized by the Method of Lines because of dense continuity Jacobians and Hessian blocks. Here the dimension of the differential states becomes dominant.

The second algorithm is characterized by additionally using the continuity and initial conditions on the differential states in $G_{1,k}$ instead of $G_{2,k}$. The continuity conditions in $G_{1,k}$ leads to the fact that the Hessian is not block diagonal and that the reduced Jacobians $\nabla G_{2,k}^T S_k^N$ and $\nabla H_k^T S_k^N$ have entries corresponding to variables on nodes before the constraints are defined. In case of e.g. fixed initial values for the differential state the dimension of the reduced QP (10) is independent from the model dimension. This algorithm was developed by Schäfer et al. 2003 [12] and implemented in the code MSOPT. In the context of Parameter Estimation this method was developed and implemented by Schlöder 1988 [13].

Both algorithms intertwine the projection and gradient evaluation steps in the reduction process so that it can be shown that it suffices to compute a reduced number of *directional* derivatives for one (reduced) SQP step:

$$\begin{array}{ccc}
 \underbrace{n(n_{xd} + n_{xa} + n_u + n_p)}_{\text{MUSCOD-II, full-space}} & \rightarrow & \underbrace{n(n_{xd} + n_u + n_p + 1)}_{\text{MUSCOD-II, reduced space}} \\
 & \rightarrow & \underbrace{\frac{1}{2}n(n+1)n_u + n n_p + n}_{\text{MSOPT}}
 \end{array}$$

using the following abbreviations for the dimensions: n number of multiple shooting intervals, n_{xd} number of differential variables, n_{xa} number of algebraic variables, n_u number of controls, n_p number of global parameters.

In order to calculate the directional differential state derivatives the BDF integrator DAESOL is used, see Bauer 2000 [1]. The derivatives are calculated as the solution of the directional Variational Differential Algebraic Equations (VDAE). In case of MUSCOD-II the linearized discretized BDF equations of the directional VDAE are *directly* solved around the solution of the nominal trajectory. Here the iteration matrix has to be computed in every BDF step. For a high number of directions this approach is advantageous and shows good performance in practice. In case of MSOPT the discretized BDF equations of the directional VDAE are solved *iteratively* together with discretized BDF equations of the nominal trajectory. Here the same iteration matrix is used. This approach conforms to the principle of Internal Numerical Differentiation (IND), see Bock 1981 [2]. The main idea is to differentiate the integrator scheme for calculation of the nominal trajectory. As a result the directional derivatives are computed efficiently and with high accuracy!

For the BDF integrator the model Jacobians are evaluated. It is necessary to do this very fast with the required accuracy. This is done with the algorithmic differentiation tool Adol-C (see Griewank et al. 1996 [10]). Since most model Jacobians are in general sparse a seedmatrix compression is used similar to the one proposed by Curtis et al. 1974 [7] to speed up computation time. The sparsity structure of the model Jacobians has to be computed only once for each model stage and is then stored.

4 A Distillation Column as a Benchmark Problem

As an application we consider the control of a high purity binary distillation column. The column is used to separate a binary mixture of Methanol and n-Propanol and has 40 bubble cup trays. An electric heating is used in the reboiler, the overhead vapor is totally condensed in a water cooled condensor. The preheated feed stream enters the column at the feed tray as saturated liquid, feedstream can be generated in arbitrary mixtures. Control variables u are the heat input in the reboiler and the reflux flow rate. The model consists of 84 differential states x (concentrations, molar holdups) and 122 algebraic states z (liquid and vapor fluxes, temperatures) and is described in Diehl (2001 [8]) on two model stages which conform to (1b). In order to generate high purity specifications we punish the deviation of the temperatures at trays 14 and 28 from reference temperatures (and add a regularization term)

$$\min_{u(\cdot), x(\cdot), z(\cdot)} \int_{t_0}^{t_0+T_p} \left\{ \left\| \tilde{T}z(t) - \tilde{T}_{\text{ref}} \right\|_2^2 + \left\| R(u(t) - u_S) \right\|_2^2 \right\} dt \quad (11a)$$

subject to the hydrodynamic model equations leading to an index one DAE system of the form

$$\left. \begin{aligned} \dot{x}(t) &= f(x(t), z(t), u(t), p) \\ 0 &= g(x(t), z(t), u(t), p), \end{aligned} \right\}, \quad t \in [t_0, t_0 + T_p], \quad (11b)$$

control conditions

$$u(t) = u_s, \quad t \in [t_0 + T_c, t_0 + T_p], \quad (11c)$$

initial conditions

$$x(t_0) = x_0, \quad (11d)$$

and path constraints

$$\begin{pmatrix} \text{molar outflow out of the reboiler} \\ \text{molar outflow out of the condenser} \end{pmatrix} \geq 0, \quad t \in [t_0, t_0 + T_p]. \quad (11e)$$

Fixed initial states are given so that the system is *not* in steady state. A Gauss-Newton variant is used for this type of problem. This optimal control problem is solved by MUSCOD-II and MSOPT in 7 iterations. In table 1 the computing times for one RSQP iteration of both algorithms on a 2.8 Ghz Intel Xeon processor under Linux 2.4.19 and GCC 3.2 are presented.

time [sec]	MUSCOD-II	MSOPT
integration	0.95	0.94
derivative generation	23.76	6.42
constraint reduction	0.32	0.13
solution of reduced QP	0.03	< 0.01

Table 1. Computing times for one RSQP iteration

Most of the time for calculation of one RSQP step is needed for the derivative generation. It is drastically reduced (here factor 4) since in the case of MSOPT we have to calculate 63 directional derivatives in contrast to 609 directional derivatives of MUSCOD-II (both with 7 multiple shooting intervals). The computation times for MUSCOD-II can be sped up by parallel calculation of the directional derivatives.

5 Conclusions and Outlook

We have presented two numerical methods for the optimization of constrained large scale nonlinear processes. We have shown practical applicability to a real world distillation column, an example with few controls, few multiple shooting intervals and fixed initial values.

The new algorithm implemented in the code MSOPT shows to be advantageous for these large scale optimization problems with not too many multiple shooting nodes, fixed initial values and a moderate number of controls.

Extensions of MSOPT for the online case are planned. Especially adaptation of the initial value embedding technique are promising (see Diehl 2001 [8]).

References

1. Bauer, I., Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsverfahren in Chemie und Verfahrenstechnik. *Ph.D. thesis*, University of Heidelberg (2000).
2. Bock, H. G., Numerical treatment of inverse problems in chemical reaction kinetics. In K. H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of Springer Series in Chemical Physics 18, 102–125, Heidelberg (1981).
3. Bock, H. G., Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, volume 2 of *Progress in Scientific Computing*, 95–121, Birkhäuser, Boston (1983).
4. Bock, H. G., and K. J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings of the 9th IFAC World Congress, Budapest*. Pergamon Press (1984).
5. Bock, H. G., Randwertproblemmethoden zur Parameteridentifizierungen in Systemen nichtlinearer Differentialgleichungen. *Bonner Mathematische Schriften*, 183 (1987).
6. Bock, H. G., E. Eich, and J. P. Schlöder, Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In K. Strehmel (ed.), *Numerical Treatment of Differential Equations*. Teubner, Leipzig (1988).
7. Curtis, A. R., M. J. D. Powell, and J. K. Reid, On the Estimation of Sparse Jacobian Matrices. *J. Inst. Maths Applies* 13, 117–119 (1974).
8. Diehl, M., Real-time optimization for large scale nonlinear processes, *Ph.D. thesis*, University of Heidelberg (2001).
9. Gabay, D., Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization. *Math. Progr. Study* 16, 18–44 (1982).
10. Griewank A., D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther: ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++; this is the updated version of the paper published in ACM TOMS, vol. 22(2), June 1996, pp. 131-167, Algor. 755, (1996).
11. Leineweber, D. B., Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models. *Fortschritt-Berichte VDI*, Reihe 3, Nr. 613 (ISBN 3-18-361303-4). VDI Verlag GmbH, Düsseldorf (1999).
12. Schäfer, A., H. G. Bock, and J. P. Schlöder, Ein effizientes reduziertes Gauss-Newton und SQP-Verfahren für die Parameterschätzung und Optimale Steuerung von chemischen Prozessen beschrieben durch DAEs mit wenig Freiheitsgraden. *Internal Report*, University of Heidelberg (2003).
13. Schlöder, J. P., Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung. *Bonner Mathematische Schriften* 187, University of Bonn (1988).
14. Schulz, V. H., Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots. *Ph.D. thesis*, University of Heidelberg (1996).

Proofs of Unsatisfiability Via Semidefinite Programming

Miguel F. Anjos

Operational Research Group, School of Mathematics, University of Southampton,
Southampton SO17 1BJ, United Kingdom. (Email: anjos@stanfordalumni.org)

Abstract. The satisfiability (SAT) problem is a central problem in mathematical logic, computing theory, and artificial intelligence. An instance of SAT is specified by a set of boolean variables and a propositional formula in conjunctive normal form. Given such an instance, the SAT problem asks whether there is a truth assignment to the variables such that the formula is satisfied. It is well known that SAT is in general NP-complete, although several important special cases can be solved in polynomial time. Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a matrix variable X is maximized (or minimized) subject to linear constraints on the elements of X and the additional constraint that X be positive semidefinite. We focus on the application of SDP to obtain proofs of unsatisfiability. Using a new SDP relaxation for SAT, we obtain proofs of unsatisfiability for some hard instances with up to 260 variables and over 400 clauses. In particular, we can prove the unsatisfiability of the smallest unsatisfiable instance that remained unsolved during the SAT Competition 2003. This shows that the SDP relaxation is competitive with the top solvers in the competition, and that this technique has the potential to complement existing techniques for SAT.

1 Introduction

The satisfiability (SAT) problem is a central problem in mathematical logic, computing theory, and artificial intelligence. An instance of SAT is specified by a set of boolean variables and a propositional formula in conjunctive normal form. Given such an instance, the SAT problem asks whether there is a truth assignment to the variables such that the formula is satisfied. It is well known that SAT is in general NP-complete, although several important special cases can be solved in polynomial time. There has been great interest in the design of efficient algorithms to solve the SAT problem; see [9] for an extensive survey.

Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a matrix variable X is maximized (or minimized) subject to linear constraints on the elements of X and the additional constraint that X be positive semidefinite. A variety of polynomial-time interior-point algorithms for solving SDPs have been proposed in the literature, and several excellent solvers for SDP are now available. We refer the reader to the recent handbook [18] for a thorough coverage of the theory

and algorithms in this area, as well as several application areas where SDP researchers have made significant contributions. We note that SDP has been very successfully applied in the development of approximation algorithms for hard combinatorial optimization problems. The survey paper [12] provides an excellent overview of the results in this area.

We are interested in the application of SDP to the basic SAT problem, and in particular in how SDP can be used to prove unsatisfiability. In [7,8] de Klerk et al. introduced an SDP relaxation for SAT, the Gap relaxation (defined below), which characterizes unsatisfiability for some interesting classes of SAT problems, such as mutilated chessboard and pigeonhole instances, as well as for 2-SAT. However, the Gap relaxation cannot detect unsatisfiability when all the clauses have length three or higher. More recently, we introduced in [2] an improved SDP relaxation which can be used to prove that a given SAT formula is unsatisfiable, independently of the lengths of the clauses in the instance. This relaxation is easily defined for every instance of SAT, and it inherits all the favourable properties of the Gap relaxation. It is constructed using ideas from a “higher liftings” paradigm for constructing SDP relaxations of discrete optimization problems.

The use of liftings has been proposed in the literature within the framework of general purpose lift-and-project methods for 0-1 optimization (see for example [5,15,13]). To summarize the paradigm behind the higher semidefinite liftings, suppose that we have a discrete optimization problem on n binary variables. The SDP relaxation (or Shor relaxation) in the space of $(n+1) \times (n+1)$ symmetric matrices is called a first lifting. Note that the rows and columns of the matrix variable in this relaxation are indexed by the binary variables themselves. To obtain higher semidefinite liftings, we allow the SDP relaxations to have the rows and columns of the matrix variable indexed by *subsets* of the discrete variables in the formulation. These larger matrices can be interpreted as higher liftings, in the spirit of the second lifting proposed in [4] and its generalization independently proposed in [11,10]. A further generalization of this paradigm, which lifts an n -tuple of binary variables into the *subset algebra* of that space, has recently been investigated in [6]. Alternatively, if a 0-1 problem is viewed as a feasibility problem over an algebraic set, then SDP relaxations can be obtained using Hilbert’s positivstellensatz [14].

From a computational point of view however, all these lifting techniques suffer from the fact that the size of the SDP relaxations grows very rapidly with the number of binary variables. For example, only second liftings for Maximum-Cut problems with only up to 27 binary variables were successfully solved in [1]. By contrast, using the SDP relaxation from [2] which can be viewed as a “partial” lifting more amenable to practical computation than the complete higher liftings, we obtained proofs of unsatisfiability for some hard instances with up to 260 variables and over 400 clauses. In particular, we were able to confirm the unsatisfiability of the smallest unsatisfiable instance that

remained unsolved during the SAT Competition 2003. This shows that the SDP relaxation is competitive with the top solvers in the competition, and that this technique has the potential to complement existing techniques for SAT. These results further motivate our ongoing research towards a practical SDP-based algorithm for satisfiability.

2 The SDP Relaxation for SAT

We consider the SAT problem in conjunctive normal form (CNF). An instance of SAT is specified by a set of variables x_1, \dots, x_n and a propositional formula $\Phi = \bigwedge_{j=1}^m C_j$, with each clause C_j having the form $C_j = \bigvee_{k \in J_j} x_k \vee \bigvee_{k \in \bar{J}_j} \bar{x}_k$ where $J_j, \bar{J}_j \subseteq \{1, \dots, n\}$, $J_j \cap \bar{J}_j = \emptyset$, and \bar{x}_i denotes the negation of x_i . (We assume without loss of generality that $|J_j \cup \bar{J}_j| \geq 2$ for every clause C_j .) The SAT problem is: Given a satisfiability instance, is Φ satisfiable, i.e. is there a truth assignment to the variables x_1, \dots, x_n such that Φ evaluates to TRUE? Special instances of SAT with certain constraints on the length of the clauses are often of particular interest, both theoretically and in practice. The notation k -SAT refers to those instances of SAT for which each clause C_j satisfies $|J_j \cup \bar{J}_j| \leq k$.

de Klerk et al. [7,8] introduced the Gap relaxation for SAT which is designed for detecting unsatisfiability. Indeed, they show that it characterizes unsatisfiability for some interesting classes of SAT problems, such as mutilated chessboard and pigeonhole instances, as well as for 2-SAT problems. For clause j and $k \in J_j \cup \bar{J}_j$, define

$$s_{j,k} := \begin{cases} 1, & \text{if } k \in J_j \\ -1, & \text{if } k \in \bar{J}_j \end{cases} \quad (1)$$

Let 1 denote TRUE and -1 denote FALSE, and let $l(C_j) = |J_j \cup \bar{J}_j|$ denote the number of literals in clause C_j . The Gap relaxation for 3-SAT may be expressed as follows:

find $X \in \mathcal{S}^{n+1}$

s.t.

$$s_{j,i_1} s_{j,i_2} X_{i_1,i_2} - s_{j,i_1} X_{0,i_1} - s_{j,i_2} X_{0,i_2} + 1 = 0, \\ \text{where } \{i_1, i_2\} = J_j \cup \bar{J}_j, \text{ if } l(C_j) = 2$$

$$s_{j,i_1} s_{j,i_2} X_{i_1,i_2} + s_{j,i_1} s_{j,i_3} X_{i_1,i_3} + s_{j,i_2} s_{j,i_3} X_{i_2,i_3} - s_{j,i_1} X_{0,i_1} - s_{j,i_2} X_{0,i_2} \\ - s_{j,i_3} X_{0,i_3} \leq 0, \text{ where } \{i_1, i_2, i_3\} = J_j \cup \bar{J}_j, \text{ if } l(C_j) = 3$$

$$\text{diag}(X) = e$$

$$X \succeq 0$$

where \mathcal{S}^n denotes the space of $n \times n$ symmetric matrices, $\text{diag}(X)$ represents a vector containing the diagonal elements of X , e denotes the vector of all ones, and $X \succeq 0$ denotes that X is positive semidefinite. This relaxation is based

on the application of the elliptic approximations introduced in [17], and the linear constraints in the SDP are obtained by expanding and linearizing the elliptic approximation for each clause. Using these elliptic approximations, it is straightforward to extend the Gap relaxation to SAT instances with any number of literals in each clause; however, the resulting relaxations are always feasible in the absence of clauses of length two, and hence unhelpful for detecting unsatisfiability. Rounding schemes and approximation guarantees for the Gap relaxation, as well as its behaviour on so-called $(2 + p)$ -SAT problems, are studied in [7].

More recently, we proposed an improved SDP relaxation which is able to detect unsatisfiability independently of the length of the clauses, and inherits all the properties of the Gap relaxation. We outline here the construction and properties of that relaxation; the complete details are presented in [2]. With $s_{j,k}$ as defined in (1), we have from [2, Proposition 1] that for $l(C_j) \geq 2$, clause C_j is satisfied by $x_i = \pm 1, i \in J_j \cup \bar{J}_j$, if and only if

$$\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq J_j \cup \bar{J}_j, |T|=t} \left(\prod_{i \in T} s_{j,i} \right) \left(\prod_{i \in T} x_i \right) \right] = 1.$$

Using this condition, we formulate the problem in symmetric matrix space. Let P denote the set of nonempty sets $I \subseteq \{1, \dots, n\}$ such that the term $\prod_{i \in I} x_i$ appears in the above formulation. Also introduce new variables $x_I := \prod_{i \in I} x_i$,

for each $I \in P$; define the vector $v := (1, x_{I_1}, \dots, x_{I_p})^T$, where p denotes the cardinality of P ; and define the rank-one matrix $Y := vv^T$, whose rows and columns are indexed by $\{\emptyset\} \cup P$. By construction of Y , we have that $Y_{\emptyset, I} = x_I$ for all $I \in P$. Using these new variables, we can formulate the SAT problem as:

$$\begin{aligned} & \text{find } Y \in \mathcal{S}^{1+p} \\ & \text{s.t.} \\ & \sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq J_j \cup \bar{J}_j, |T|=t} \left(\prod_{i \in T} s_{j,i} \right) Y_{\emptyset, T} \right] = 1, \quad j = 1, \dots, m \\ & \text{diag}(Y) = e \\ & \text{rank}(Y) = 1 \\ & Y \succeq 0. \end{aligned}$$

Relaxing this formulation by omitting the rank constraint would give an SDP relaxation for SAT. However, in order to improve the SDP relaxation, we first add *redundant* constraints to this formulation. This approach of adding redundant constraints so as to tighten the resulting SDP relaxation is discussed in detail in [3].

The constraint $\text{rank}(Y) = 1$ implies that for every triple I_1, I_2, I_3 of subsets of indices in P such that the symmetric difference of any two equals

the third, the following three equations hold:

$$Y_{\emptyset, I_1} = Y_{I_2, I_3}, \quad Y_{\emptyset, I_2} = Y_{I_1, I_3}, \quad \text{and} \quad Y_{\emptyset, I_3} = Y_{I_1, I_2}. \quad (2)$$

Hence we can add some or all of these redundant constraints to the formulation (without affecting its validity). We add to the formulation above the equations of the form (2) for all the triples $\{I_1, I_2, I_3\} \subseteq P$ satisfying the symmetric difference condition and such that $(I_1 \cup I_2 \cup I_3) \subseteq (J_j \cup \bar{J}_j)$ for some clause j . Our final formulation of the SAT problem is thus:

find $Y \in \mathcal{S}^{1+p}$

s.t.

$$\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq J_j \cup \bar{J}_j, |T|=t} \left(\prod_{i \in T} s_{j,i} \right) Y_{\emptyset, T} \right] = 1, \quad j = 1, \dots, m$$

$$Y_{\emptyset, I_1} = Y_{I_2, I_3}, \quad Y_{\emptyset, I_2} = Y_{I_1, I_3}, \quad \text{and} \quad Y_{\emptyset, I_3} = Y_{I_1, I_2}, \quad \forall \{I_1, I_2, I_3\} \subseteq P$$

such that $I_1 \Delta I_2 = I_3$ and $(I_1 \cup I_2 \cup I_3) \subseteq (J_j \cup \bar{J}_j)$ for some j

$$\text{diag}(Y) = e$$

$$\text{rank}(Y) = 1$$

$$Y \succeq 0$$

where $I_i \Delta I_j$ denotes the symmetric difference of I_i and I_j . Removing the rank constraint yields an SDP relaxation with the following properties:

Theorem 1 [2, Theorem 2] *Given a SAT propositional formula in CNF, the following statements hold for the semidefinite relaxation above:*

- *If the SDP is infeasible, then the formula is unsatisfiable.*
- *If the SDP is feasible, and Y is a feasible matrix such that $\text{rank } Y \leq 3$, then a truth assignment satisfying the formula can be obtained from Y . Hence the formula is satisfiable.*

Hence, to prove that a SAT instance is unsatisfiable, it suffices to verify that the SDP relaxation is infeasible (when that is indeed the case). We now present computational results that illustrate the potential of our SDP relaxation for proving unsatisfiability.

3 Some Proofs of Unsatisfiability Via SDP

Researchers in SDP have developed a variety of excellent solvers, most of which are freely available. An extensive listing of solvers is available at http://www-user.tu-chemnitz.de/~helmberg/sdp_software.html. For the application to SAT, it is important to use a solver which, when given an infeasible SDP, provides us with a certificate of infeasibility, because that certificate is for us a proof of unsatisfiability for the SAT instance. We used the solver SDPT3 (version 3.0) [16] with its default settings (available at

Table 1. Results for the Nine hgen8 Instances from SAT Competition 2003

Problem	# of variables	# of clauses	Size of Y	# of SDP constraints	SDP is infeasible	Total CPU seconds	Solved at SAT 2003
n120-01	120	197	542	3862	Yes	704	Yes
n120-02	120	193	537	3838	Yes	611	Yes
n120-03	120	193	539	3846	Yes	583	Yes
n180-01	180	279	793	5668	Yes	2194	Yes
n180-02	180	279	793	5668	Yes	2142	Yes
n180-03	180	280	791	5661	Yes	2188	Yes
n260-01	260	391	1132	8096	Yes	6938	No
n260-02	260	404	1143	8153	Yes	7455	No
n260-03	260	399	1134	8112	Yes	7678	No

n120-01 denotes problem `hgen8-n120-01-S563767109.shuffled-as.sat03-875`
 n120-02 denotes problem `hgen8-n120-02-S1654058060.shuffled-as.sat03-876`
 n120-03 denotes problem `hgen8-n120-03-S1962183220.shuffled-as.sat03-877`
 n180-01 denotes problem `hgen8-n180-01-S1524349002.shuffled-as.sat03-880`
 n180-02 denotes problem `hgen8-n180-02-S1125510326.shuffled-as.sat03-881`
 n180-03 denotes problem `hgen8-n180-03-S1436192352.shuffled-as.sat03-882`
 n260-01 denotes problem `hgen8-n260-01-S1597732451.shuffled-as.sat03-885`
 n260-02 denotes problem `hgen8-n260-02-S1396509323.shuffled-as.sat03-886`
 n260-03 denotes problem `hgen8-n260-03-S722413478.shuffled-as.sat03-887`

Table 2. Results for the Additional hgen8 Instances

Problem	# of variables	# of clauses	Size of Y	# of SDP constraints	SDP is infeasible	Total CPU seconds
200-01	200	309	881	6290	Yes	2821
200-02	200	314	882	6299	Yes	3243
200-03	200	306	877	6271	Yes	2530
220-01	220	339	966	6900	Yes	3843
220-02	220	344	969	6917	Yes	4279
220-03	220	341	967	6906	Yes	4757
240-01	240	362	1044	7475	Yes	5455
240-02	240	365	1047	7490	Yes	5563
240-03	240	366	1046	7487	Yes	5376

<http://www.math.nus.edu.sg/~mattohk/sdpt3.html>) and running on a 2.4GHz Pentium IV with 1.5Gb of RAM.

The results we present are for randomly generated SAT instances obtained using the generator hgen8. The source code to generate these instances, which includes an explanation of their structure, is available at <http://logic.pdmi.ras.ru/~hirsch/benchmarks/hgen8.html>. A set of 12 instances generated using hgen8 was submitted for the SAT competition 2003 (see <http://www.satlive.org/SATCompetition/2003/index.jsp>). We verified that our SDP relaxation is infeasible for nine of these instances; these results are presented in Table 1. (The SDP relaxations for the remaining instances in the set were too large for the computing resources available.) In particular, the SDP relaxation was able to prove the unsatisfiability of the instance n260-01 in Table 1, which was the smallest unsatisfiable instance that remained unsolved during the competition. (An instance remained unsolved if none of the top five solvers was able to solve it in two hours, running on an Athlon 1800+ with 1Gb of RAM.) Indeed, the SDP relaxation appears to be quite effective on the type of instances generated by hgen8, as we randomly generated several more instances of varying sizes and all the corresponding SDP relaxations were infeasible. The additional results are presented in Table 2. It would be interesting to relate the structure of these instances to the success of the SDP relaxation.

To conclude, our main point here is that for some instances of SAT, the SDP relaxation is competitive with the top solvers in the SAT 2003 competition, and therefore the SDP approach has the potential to complement existing techniques for SAT. These results are an encouraging step in our ongoing research towards a practical SDP-based algorithm for satisfiability.

References

1. M.F. Anjos. *New Convex Relaxations for the Maximum Cut and VLSI Layout Problems*. PhD thesis, University of Waterloo, 2001. Published online at <http://etd.uwaterloo.ca/etd/manjos2001.pdf>.
2. M.F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. *Math. Program.* (Ser. A), to appear, 2004.
3. M.F. Anjos and H. Wolkowicz. Semidefinite programming for discrete optimization and matrix completion problems. *Discrete Appl. Math.*, 123(1–2):513–577, 2002.
4. M.F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. *Discrete Appl. Math.*, 119(1–2):79–106, 2002.
5. E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, 58(3, Ser. A):295–324, 1993.
6. D. Bienstock and M. Zuckerberg. Subset algebra lift operators for 0-1 integer programming. Technical Report CORC 2002-01, Columbia University, July 2002.

7. E. de Klerk and H. Van Maaren. On semidefinite programming relaxations of $(2 + p)$ -SAT. *Ann. Math. Artif. Intell.*, 37(3):285–305, 2003.
8. E. de Klerk, H. Van Maaren, and J.P. Warners. Relaxations of the satisfiability problem using semidefinite programming. *J. Automat. Reason.*, 24(1-2):37–65, 2000.
9. J. Gu, P.W. Purdom, J. Franco, and B.W. Wah. Algorithms for the satisfiability (SAT) problem: a survey. In: *Satisfiability problem: theory and applications (Piscataway, NJ, 1996)*, pages 19–151. Amer. Math. Soc., Providence, RI, 1997.
10. J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817 (electronic), 2000/01.
11. J.B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769 (electronic), 2002.
12. M. Laurent and F. Rendl. Semidefinite programming and integer programming. In: G. Nemhauser K. Aardal and R. Weismantel, editors, *Handbook on discrete optimization*. to appear.
13. L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991.
14. P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2, Ser. B):293–320, 2003.
15. H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.
16. K.C. Toh, M.J. Todd, and R.H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11/12(1-4):545–581, 1999.
17. H. van Maaren. Elliptic approximations of propositional formulae. *Discrete Appl. Math.*, 96/97:223–244, 1999.
18. H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming*. Kluwer Academic Publishers, Boston, MA, 2000.

Algorithms with Performance Guarantees for a Metric Problem of Finding Two Edge-Disjoint Hamiltonian Circuits of Minimum Total Weight *

Alexey E. Baburin, Edward Kh. Gimadi, and Natalie M. Korkishko

Sobolev Institute of Mathematics, prospekt Akademika Koptyuga 4, 630090
Novosibirsk, Russia

Abstract. This paper aims at describing a metric problem of finding two minimum total weight edge-disjoint Hamiltonian circuits in a graph with two weight functions. The problem is *NP*-hard in strong sense if the weight functions w_1 and w_2 are different or equal. We construct two approximation $O(n^3)$ algorithms whose worst-case performance guarantees asymptotically equal $12/5$ (in case of the different functions), and $9/4$ (when w_1 and w_2 are equal).

1 Introduction

Let $G = (V, E)$ be a complete undirected graph with n vertices. The edges of the graph are weighted with two functions $w_1 : E \rightarrow R$, $w_2 : E \rightarrow R$. It is supposed that the triangle inequality holds

$$w_1(i, j) \leq w_1(i, k) + w_1(j, k), w_2(i, j) \leq w_2(i, k) + w_2(j, k)$$

for each three vertices $i, j, k \in V$. By $W_i(H)$ we denote $W_i(H) = \sum_{e \in H} w_i(e)$, where $i = 1, 2$. We consider a problem of finding two edge-disjoint Hamiltonian circuits $H_1 \subset E$ and $H_2 \subset E$ such that their total weight $W_1(H_1) + W_2(H_2)$ is minimal. The problem is *NP*-hard in strong sense if the weight functions w_1 and w_2 are different or equal.

2 Complexity of the Problem

It can be easily seen that the problem is *NP*-hard in general case. For any minimum TSP with a weight function w of edges let $w_1 \equiv w$, $w_2 \equiv 0$. For these weight functions w_1 and w_2 optimality of a solution of our problem gives optimality of one Hamiltonian circuit from the solution for the minimum TSP.

But we are interested also in the case of equal weight functions $w_1 \equiv w_2$, thus let us proceed to the proof of its *NP*-hardness.

* This research was supported by the Russian Foundation for Basic Research (grant 02-01-01153), Project of Russian Federation "Nauchnaia Shkola-313.203.1", and INTAS (grant 00-217)

We need to show that the problem of finding two edge-disjoint Hamiltonian circuits in a graph is NP -hard. For that we will reduce the Hamiltonian circuit problem to this one.

A simple graph $G = (V, E)$, $V = \{v_1, \dots, v_n\}$ is given. It is required to find a Hamiltonian circuit H in it or to show that it does not exist.

We will refer to the case of even n . We construct a graph \tilde{G} that corresponds to the given graph G . \tilde{G} consists of two copies of G connected by n four-vertex cliques K_1, \dots, K_n .

An example of such a construction is shown on Fig. 1.

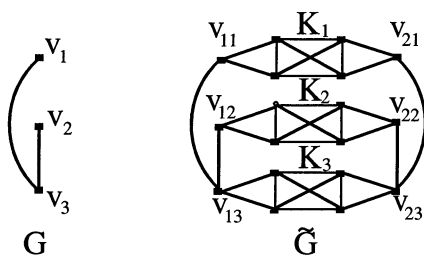


Fig. 1.

On the graph \tilde{G} we solve a problem of finding two edge-disjoint Hamiltonian circuits.

Lemma 1. *If there is a Hamiltonian circuit in G , then there are two edge-disjoint Hamiltonian circuits in \tilde{G} .*

Proof. Suppose we have a Hamiltonian circuit H in G . The cycle can be decomposed into two perfect matchings. For the first Hamiltonian circuit in \tilde{G} we take the first matching in the first copy of G in \tilde{G} , the second matching in the second copy of G in \tilde{G} and n connecting paths that connect corresponding copies of vertices from G in \tilde{G} . The second Hamiltonian circuit is constructed symmetrically, although the connecting paths are chosen differently in order to make these two Hamiltonian circuits edge-disjoint.

Lemma 2. *If there are two edge-disjoint Hamiltonian circuits in \tilde{G} , then there is a Hamiltonian circuit in G .*

Proof. Suppose we have two Hamiltonian circuits in \tilde{G} . Let one of this Hamiltonian circuits be H . If we unite pairs of copies of vertices from G in \tilde{G} and get rid of the connecting constructions between them, then the Hamiltonian circuit H transforms into a Hamiltonian circuit in G .

Lemmas 1 and 2 show that the problem of finding a Hamiltonian circuit in a graph with even number of vertices is polynomially reducible to the

problem of finding two edge-disjoint Hamiltonian circuits in a graph. The last problem is from NP class, and the first is NP -complete. We conclude that the last problem is NP -complete.

3 Algorithm 1 for the Metric Problem with Equal Weight Functions

Let $w_1 = w_2 = w$. We want to find two edge-disjoint Hamiltonian circuits $H_1 \subset E$, $H_2 \subset E$, such that $W(H_1) + W(H_2)$ is minimum. Let the minimum total weight of these Hamiltonian circuits be W^* .

Stage 1. We find the first Hamiltonian circuit $H_1 \subset G$ using Christofides's and Serdyukov's algorithm [1,2]. Assume $H_1 = \{1, 2, \dots, n, 1\}$.

Stage 2. If n is odd, then $H_2 = \{1, 3, 5, \dots, n, 2, 4, \dots, n-1, 1\}$ and Algorithm 1 has finished. Else Algorithm 1 performs stage 3.

Stage 3. (The stage 3 is performed if n is even.) Let C_2 (pro tanto C_3) be a totality of the n different edges like $(i, i+2)$ (pro tanto $(i, i+3)$). Obviously, $C_2 \subset G$ is a 2-factor which consists of two cycles $C' = \{1, 3, \dots, n-1, 1\}$ and $C'' = \{2, 4, \dots, n-2, n, 2\}$. The former cycle includes all odd nodes, the latter cycle includes all even nodes.

Description of Algorithm 1 is complete.

Consider a system $\{H^1, \dots, H^m\}$ of $m = n/2$ Hamiltonian circuits (see Fig. 2).

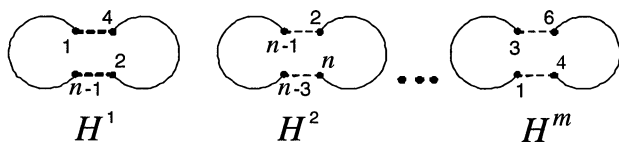


Fig. 2.

The Hamiltonian circuit H^j is obtained by splicing the cycles C' and C'' . Recall that $C_2 = C' \cup C''$ is 2-factor. We delete two edges $(2(m-j)+1, 2(m-j)+3) \in C'$ and $(2(m-j)+4, 2(m-j)+6) \in C''$ from C_2 and add two new edges $(2(m-j)+1, 2(m-j)+4), (2(m-j)+3, 2(m-j)+6) \in C_3$ (see Fig. 3).

We choose H_2 as the minimum weight Hamiltonian circuit from the system $\{H^1, \dots, H^m\}$.

Theorem 1. *If the weight functions are equal, then the metric problem of finding two edge-disjoint Hamiltonian circuits can be solved by Algorithm 1 in time $O(n^3)$ with the following performance guarantee*

$$\Delta = \begin{cases} 9/4, & \text{if } n \text{ is odd;} \\ 9/4 + 3/n, & \text{if } n \text{ is even.} \end{cases}$$

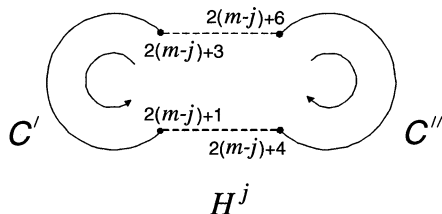


Fig. 3.

Proof. CASE 1: n is odd. Let us denote by W_{TSP}^* the minimal of the Hamiltonian circuit's weights. Obviously, that $2W_{TSP} \leq W^*$. According to the algorithm from [1,2]

$$W(H_1) \leq (3/2)W_{TSP}^*.$$

We have $W(H_2) \leq 2W(H_1)$ by the triangle inequality. Using the previous inequalities we have

$$\Delta = \frac{W(H_1) + W(H_2)}{W^*} \leq \frac{3W(H_1)}{2W_{TSP}^*} \leq \frac{3(3/2)W_{TSP}^*}{2W_{TSP}^*} = 9/4.$$

CASE 2: n is even. Note that a total weight of all Hamiltonian circuits H^1, \dots, H^m is equal to $(m-1)W(C_2) + 2W(C_3)$. Hence the minimum weight of a Hamiltonian circuit from the set satisfies the inequality

$$W(H_2) \leq \frac{(m-1)W(C_2) + 2W(C_3)}{m}.$$

Since the triangle inequality we have

$$W(C_2) \leq 2W(H_1), \quad W(C_3) \leq 3W(H_1).$$

Hence

$$W(H_2) \leq 2W(H_1)(1 + 2/m).$$

Taking into account the previous inequalities we have the performance guarantee in case n is even

$$\begin{aligned} \Delta &= \frac{W(H_1) + W(H_2)}{W^*} \leq \frac{W(H_1) + 2W(H_1)(1 + 2/m)}{2W_{TSP}^*} \\ &= \frac{W(H_1)(3 + 2/m)}{2W_{TSP}^*} \leq \frac{(3/2)W_{TSP}^*(3 + 2/m)}{2W_{TSP}^*} = 9/4(1 + 4/(3n)). \end{aligned}$$

4 Algorithm 2 for the Metric Problem with Different Weight Functions

Let $G_1 = (V_1, E_1) = G = (V, E)$ with the weight function w_1 , $G_2 = (V_2, E_2) = G = (V, E)$ with the weight function w_2 .

Stage 1. We find two Hamiltonian circuits $H_1 \subset G_1$ and $H_2 \subset G_2$ using the Christofides's and Serdyukov's algorithm [1,2]. Their weights satisfy $W_1(H_1) \leq (3/2)W_1(H_1^*)$ and $W_2(H_2) \leq (3/2)W_2(H_2^*)$, where H_1^* and H_2^* are Hamiltonian circuits of minimum weight in G_1 and G_2 . Clearly, that $W_1(H_1^*) + W_2(H_2^*) \leq W^*$. If H_1^* and H_2^* are edge-disjoint, then Algorithm 2 has finished. Else Algorithm 2 performs Stage 2.

Stage 2. We take H_1 as the first Hamiltonian circuit. The aim of the stage 2 is to convert H_2 into the Hamiltonian circuit \tilde{H}_2 which is edge-disjoint with H_1 . First of all we renumber the edges of G according to H_2 .

Let $n = 5m + r$, $0 \leq r < 5$. Let S^k be a set of chains (intervals) P_i^1, \dots, P_i^m which consecutively lie in H_2 and are obtained by dividing H_2 into $m - r$ parts that consist of 5 linked edges, and into r parts that consist of 6 linked edges. The index k means that the first chain P_1^k begins with the node $k = 1, 2, \dots, K$ where

$$K = \begin{cases} 5, & \text{if } n \text{ is divisible by } 5; \\ n, & \text{if } n \text{ isn't divisible by } 5. \end{cases}$$

Let denote by e'_{ik} and e''_{ik} the terminating edges of P_i^k .

We organize the sets S^1, \dots, S^K of the chains described above. The nodes of every chain $P \in S^k$ are detoured on the path \tilde{P} which possess the following properties:

- 1°. \tilde{P} starts with the first node of the chain;
- 2°. \tilde{P} passes every inner node only once;
- 3°. \tilde{P} finishes at the end node;
- 4°. \tilde{P} does not intersect the edges of H_1 .

(Later, Lemma 3 shows the correctness of finding the path \tilde{P} that consists of 5 or 6 edges).

We splice all paths found for the set S^k , and obtain a Hamiltonian circuit H^k which doesn't intersect with the edges of H_1 . We choose \tilde{H}_2 as a solution where \tilde{H}_2 is minimum weight Hamiltonian circuit from H^1, \dots, H^K .

Description of Algorithm 2 is complete.

Further, we assume $w = w_2$.

Lemma 3. *Let the chain $P \in H_2$ consist of 5 or 6 consecutively joined edges. We can construct the path \tilde{P} (which possesses the properties 1°–4°) for a constant number of atomic operations. And \tilde{P} has total weight*

$$W(\tilde{P}) \leq 3 \cdot W(P) - 2(w(e') + w(e'')),$$

where e' and e'' are the first (starting) and the last (finishing) edges of the chain.

The proof is omitted.

Lemma 4. *The length of the Hamiltonian circuit \widetilde{H}_2 satisfies the condition*

$$W(\widetilde{H}_2) \leq \begin{cases} (11/5)W(H_2), & \text{if } n \text{ is divisible by } 5; \\ (11/5)(1 + 16/(11n))W(H_2), & \text{if } n \text{ is not divisible by } 5. \end{cases}$$

Proof. Since Lemma 3 for the lengths of Hamiltonian circuits H^1, \dots, H^K we have

$$\begin{aligned} W(H^k) &= \sum_{i=1}^m W(\widetilde{P}_i^k) \leq 3 \sum_{i=1}^m W(P_i^k) - 2 \sum_{i=1}^m (w(e'_{ik}) + w(e''_{ik})) \\ &\leq 3W(H_2) - 2 \sum_{i=1}^m (w(e'_{ik}) + w(e''_{ik})). \end{aligned}$$

For the length of the Hamiltonian circuit H_2 we get

$$W(H_2) = w(n, 1) + \sum_{j=1}^{n-1} w(j, j+1).$$

As the Hamiltonian circuit \widetilde{H}_2 has the least weight of Hamiltonian circuits H^1, \dots, H^K we conclude that

$$W(\widetilde{H}_2) \leq (1/K) \sum_{k=1}^K W(H^k) \leq 3W(H_2) - (2/K) \sum_{k=1}^K \sum_{i=1}^m (w(e'_{ik}) + w(e''_{ik})).$$

If n is divisible by 5, we have

$$W(\widetilde{H}_2) \leq (1/5) \sum_{k=1}^5 W(H^k) \leq 3W(H_2) - (2/5) \sum_{k=1}^5 \sum_{i=1}^m (w(e'_{ik}) + w(e''_{ik})).$$



Fig. 4.

On the Fig. 4 gray and black colors represent (for each 5-edge's chain) first and last edges respectively. We subtract these edges namely.

Considering that each edge e of the Hamiltonian circuit H_2 is added in the sum exactly twice we get

$$W(\widetilde{H}_2) \leq 3W(H_2) - (2/5)2W(H_2) = (11/5)W(H_2).$$

If n is not divisible by 5, we have

$$W(\tilde{H}_2) \leq (1/n) \sum_{k=1}^n W(H^k) \leq 3W(H_2) - (2/n) \sum_{k=1}^n \sum_{i=1}^m (w(e_{i1}^k) + w(e_{i2}^k)).$$

Since each edge e of the Hamiltonian circuit H_2 is added in the sum exactly $2m$ times we get

$$W(\tilde{H}_2) \leq 3W(H_2) - (2/n)2mW(H_2).$$

Using an inequality $5m \geq n - 4$ we conclude

$$W(\tilde{H}_2) \leq (11/5)(1 + 16/(11n))W(H_2).$$

The lemma is proven.

Theorem 2. *If the weight functions are different, then the metric problem of finding two edge-disjoint Hamiltonian circuits can be solved by Algorithm 2 in time $O(n^3)$ with the following performance guarantee*

$$\Delta \leq \begin{cases} (12/5), & \text{if } n \text{ is divisible by 5;} \\ (12/5)(1 + 1/n), & \text{if } n \text{ is not divisible by 5.} \end{cases}$$

Proof. First of all we find the performance guarantee.

CASE 1: n is divisible by 5. From the inequalities

$$\begin{aligned} W_1(H_1^*) + W(H_2^*) &\leq W^* \leq W_1(H_1) + W(\tilde{H}_2) \\ &\leq W_1(H_1) + (11/5)W(H_2) \leq (3/2)(W_1(H_1^*) + (11/5)W_1(H_2^*)) \end{aligned}$$

we have

$$\Delta = \frac{W_1(H_1) + W(\tilde{H}_2)}{W^*} \leq \frac{3}{2} \frac{W_1(H_1^*) + \frac{11}{5}W(H_2^*)}{W_1(H_1^*) + W(H_2^*)} = \frac{3}{2} \left(1 + \frac{6/5}{1+t}\right),$$

where $t = W_1(H_1^*)/W(H_2^*)$.

Exchanging the roles of G_1 and G_2 we obtain the following inequality

$$\Delta \leq \frac{3}{2} \left(1 + \frac{6/5}{1+1/t}\right).$$

From here we have the performance guarantee of Algorithm 2

$$\Delta \leq \frac{3}{2} \left(1 + \min \left\{ \frac{6/5}{1+t}, \frac{6/5}{1+1/t} \right\} \right) \leq (3/2)(1 + 3/5) = 12/5.$$

CASE 2: n is not divisible by 5. Since Lemma 4 we have

$$\Delta = \frac{W_1(H_1) + W(\tilde{H}_2)}{W^*} \leq (3/2) \frac{W_1(H_1^*) + (11/5)(1 + 16/(11n))W(H_2^*)}{W_1(H_1^*) + W(H_2^*)}$$

$$= (3/2) \left(1 + \frac{6/5 + 16/(5n)}{1 + t} \right),$$

where $t = W_1(H_1^*)/W(H_2^*)$.

Exchanging the roles of G_1 and G_2 we obtain the following inequality

$$\Delta \leq (3/2) \left(1 + \frac{6/5 + 16/(5n)}{1 + 1/t} \right).$$

Consequently

$$\Delta \leq (3/2) \left(1 + \min \left\{ \frac{6/5 + 16/(5n)}{1 + t}, \frac{6/5 + 16/(5n)}{1 + 1/t} \right\} \right) \leq 12/5(1 + 1/n).$$

We can see the same performance guarantee for both cases when n is and is not divisible by 5.

The time complexity of Algorithm 2 is determined by the time complexity of finding a maximum weight matching [3] in the Christofides's and Serdyukov's algorithm [1,2], which is $O(n^3)$.

Acknowledgment. The authors are thankful to Alexander Ageev and Mikhail Paschenko for helpful discussions on the complexity of the problem.

References

1. N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem . Technical report CS-93-13, Carnegie Mellon University, 1976.
2. A. I. Serdyukov, On some extrem circuits in graphs (Russian), Upravlyaemye Sistemy. Sbornik Nauchnih Trudov. Sobolev Institute of Mathematics of SB SAS. Novosibirsk, v. 17, 1978, pp. 76-79.
3. H. Gabow, An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems // In Proceedings of the 15th annual ACM symposium on theory of computing (Boston, Apr. 25-27). ACM, New York, pp. 448-456.

A Quadratic Optimization Model for the Consolidation of Farmland by Means of Lend-Lease Agreements

Andreas Brieden and Peter Gritzmann

brieden/gritzman@ma.tum.de
Zentrum Mathematik, Technische Universität München
D-80290 München, Germany

Abstract. In many regions farmers cultivate a number of small lots that are distributed over a wider area. This leads to high overhead costs and economically prohibits use of high tech machinery hence results in a non-favorable cost-structure of production. The classical form of land consolidation is typically too expensive and too rigid, whence consolidation based on lend-lease agreements has been suggested. In order to exploit the potential of this method specific mathematical optimization algorithms are needed, particularly since the underlying problem is NP-complete even in very restricted cases.

This paper introduces a quadratic optimization model and shows its practicality for some typical regions in Northern Bavaria, Germany.

1 Introduction

In many regions in Germany and other countries individual farmers cultivate a number of rather small lots that are distributed over a wider area. Figure 1 depicts an example from Northern Bavaria, Germany.¹ On the average, today's Bavarian farmers cultivate twelve lots with an average size of 1.45 hectare ² (ha).

Naturally, in such a situation the farmers are faced with an unnecessarily large unproductive overhead, the man power and equipment needed for reaching each individual lot. In fact, according to calculations of the State Institute of Bavaria for Agriculture (LfL) in Munich, Germany, this overhead may reach as much as about ten percent of the total net income. Further, larger machinery cannot be used profitably. The reason for such small-split farmland lies in the heritage laws combined with an increase of leased but not owned agricultural property.

In the classical form of land consolidation all lots are first combined to one large region which is subsequently partitioned into the new individual properties of the corresponding farmers. At the same time the complete infrastructure of the area is replanned. This legal change of ownership is very

¹ The reproduction process did not allow for colors; the original multicolor figures can be viewed under <http://www-m9.ma.tum.de/dm/consolidation/OR2003>

² 1 ha = 10.000m² = 2,471 acres

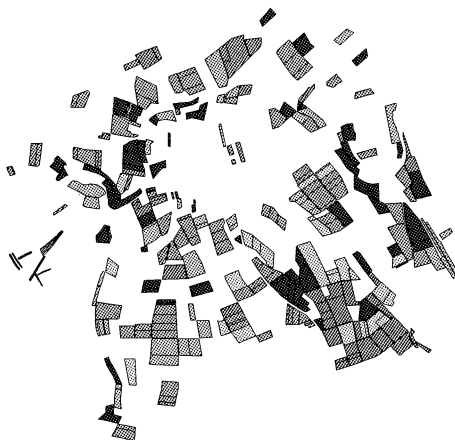


Fig. 1. Status quo distribution of the lots of individual farmers; lots depicted in the same grey value belong to the same farmer.

costly, time intense, rigid and its fairness is always up to dispute. In fact, the cost per hectare are about 2000 Euro, the duration of the procedure not rarely close to a decade, and normally each farmer is forced to participate.

Therefore the LfL suggested a lend-lease based redistribution of the *existing* lots. This means that the redistribution will not result in any resizing or legal change of ownership but only in a (temporary) reassignment of utilization. Consequently, the method is easy to implement, inexpensive and very flexible. It actually works on a voluntary basis, no farmer is forced to participate. Also individual preferences can be accommodated, and updates can be performed on any time scale that is required.

In practice, the real bottleneck of this approach is that adequate techniques for the redistribution of utilization rights that take into account the quality of the soil, the (real or potential) European Union subsidy benefits and many other relevant factors are not at hand of the farmers or the experts of the ministry. In fact, even extremely simple seeming cases can be shown to be NP-complete, see [4].

The purpose of this note is to outline a new quadratic optimization model for the lend-lease based consolidation of farming. A more detailed description and analyses will be given in [4]. In Section 2 we present the core model and Section 3 contains computational results for two regions in Northern Bavaria.

2 Mathematical Model

Let the given farmland be split into – say – l^* lots L_1, \dots, L_{l^*} , and let the function $\omega : \{L_1, \dots, L_{l^*}\} \rightarrow [0, \infty[$ associate with each lot its size (or soil quality or subsidy potential or any other kind of measure for its quality). Now, the lots are cultivated by – say – f^* different farmers F_1, \dots, F_{f^*} , i.e.,

we are given a partition (I_1, \dots, I_{f^*}) of the index set $I = \{1, \dots, l^*\}$ such that for $f = 1, \dots, f^*$ the set $P_f = \{L_l : l \in I_f\}$ describes the current property of farmer f ; of course, $\beta_f = \sum_{l \in I_f} \omega(L_l)$ is the corresponding total farm size. We introduce variables ξ_{lf} that indicate whether lot l belongs to farmer f , i.e., for the given land distribution

$$\xi_{lf} = \begin{cases} 1, & \text{if } L_l \in I_f, \\ 0, & \text{if } L_l \notin I_f. \end{cases}$$

Hence we can capture the conservation of size by means of the following system of linear equalities.

$$\sum_{l=1}^{l^*} \omega(L_l) \xi_{lf} = \beta_f, \quad f = 1, \dots, f^*$$

Of course, in practice one would prefer more flexibility and allow for some small deviation; it can be modeled with the aid of some positive relative error parameters ϵ_f , $f = 1, \dots, f^*$. Since each lot has to be assigned to precisely one farmer, we obtain the following system of constraints.

Constraints

$$\begin{aligned} \sum_{l=1}^{l^*} \omega(L_l) \xi_{lf} &\leq (1 + \epsilon_f) \beta_f, & f = 1, \dots, f^* \\ \sum_{l=1}^{l^*} \omega(L_l) \xi_{lf} &\geq (1 - \epsilon_f) \beta_f, & f = 1, \dots, f^* \\ \sum_{f=1}^{f^*} \xi_{lf} &= 1, & l = 1, \dots, l^* \\ \xi_{lf} &\in \{0, 1\}, & f = 1, \dots, f^*; l = 1, \dots, l^*. \end{aligned}$$

Of course, it is straight forward to extend this model to capture other measures of quality of the lots and to accommodate individual preferences. For instance, if some farmers insist on some of their lots being excluded from the redistribution, we simply preset the corresponding variables to 1.

Among all feasible solutions specified by the Constraints we are longing for one whose corresponding distribution is “optimal”; hence an objective function has to be fixed. Our model aims at minimizing the average distances between the lots of one farmer and maximizing the distances between lots of different farmers. More specifically, we identify with each individual farmer f the “virtual center of gravity” cg_f of all his lots and maximize the sum of the square of their mutual Euclidean distances. (A mathematical justification for this approach in terms of separation is given in [4].)

To compute cg_f , we utilize information available through the geographic information system (GIS) of the LfL. In fact, besides information on lot sizes, soil quality, etc. GIS provides for each lot L_l a tuple $g_l = (g_{l,1}, g_{l,2})^T$ of co-ordinates, the arithmetic mean of Gauß-Krüger-coordinates of the “corners” of the lots.³ Then the virtual center of gravity for farmer f is defined by

$$cg_f = \frac{1}{\beta_f} \sum_{l=1}^{l^*} \omega(L_l) g_l \xi_{lf}.$$

³ The Gauß-Krüger-coordinates are projections of points of the terrestrial sphere on a suitable tangent plane.

Note that we do not normalize by the actual farm size

$$\sum_{l=1}^{l^*} \omega(L_l) \xi_{lf}$$

but by the initial farm size β_f in order to keep the constraints linear. We are then interested in the vectors of pairwise difference

$$\delta_{f_1, f_2} = (\delta_{f_1, f_2; 1}, \delta_{f_1, f_2; 2})^T = cg_{f_1} - cg_{f_2}, \quad 1 \leq f_1 < f_2 \leq f^*$$

that altogether can be interpreted as a $f^*(f^* - 1)$ dimensional vector x whose Euclidean norm $\|x\|_2$ has to be maximized over the 0-1-vectors of a polytope. More precisely, we obtain the following optimization problem.

Optimization Problem

$$\max \sum_{f_1=1}^{f^*-1} \sum_{f_2=f_1+1}^{f^*} \sum_{i=1}^2 \delta_{f_1, f_2; i}^2$$

s.t.

$$\begin{aligned} \sum_{l=1}^{l^*} \omega(L_l) \xi_{lf} &\leq (1 + \epsilon_f) \beta_f, & f = 1, \dots, f^* \\ \sum_{l=1}^{l^*} \omega(L_l) \xi_{lf} &\geq (1 - \epsilon_f) \beta_f, & f = 1, \dots, f^* \\ \sum_{f=1}^{f^*} \xi_{lf} &= 1, & l = 1, \dots, l^* \\ \xi_{lf} &\in \{0, 1\}, & f = 1, \dots, f^*; l = 1, \dots, l^* \\ \beta_f cg_f - \sum_{l=1}^{l^*} \omega(L_l) g_l \xi_{lf} &= 0, & f = 1, \dots, f^* \\ \delta_{f_1, f_2} - (cg_{f_1} - cg_{f_2}) &= 0, & 1 \leq f_1 < f_2 \leq f^*. \end{aligned}$$

Relaxing the constraints

$$\xi_{lf} \in \{0, 1\}$$

to

$$0 \leq \xi_{lf}$$

(note that the inequalities $\xi_{lf} \leq 1$ are redundant) we obtain the problem of maximizing the (square of the) Euclidean norm over the orthogonal projection of a polytope in $((l^* + f^* + 1)f^*)$ -dimensional space onto an $(f^*(f^* - 1))$ -dimensional linear subspaces. For fixed f^* the latter can be solved with arbitrary accuracy in polynomial time [5,6]. The algorithm is based on an approximation of the $f^*(f^* - 1)$ -dimensional Euclidean unit ball by some suitable polytope. The facet hyperplanes of the approximating polytope lead to linear objective functions; hence linear programming can be applied to approximate the optimal value of the original convex quadratic maximization problem. (For a detailed study on this approach and various related complexity results see [8,7,1,3].) Here we use only a linear number of linear programs.

Note, that this relaxation ignores the integrality conditions. However, it can be shown for the Optimization Problem that all but at most $2f^*$ coordinates of any vertex of the underlying polytope that maximizes a linear objective function are already integer. This leads to at most f^* remaining lots that can then be easily post-optimized.

3 Computational Results

The algorithm sketched above is already in practical use. To illustrate its performance we present the results for two regions in Northern Bavaria, with 5 (13) participating farmers cultivating a total of 306 (861) lots with an overall size of approximately 331 (436) hectares in the first (second) region⁴. These numbers can be seen as quite typical for many other regions, particularly in Franconia and Swabia.

Recall, that our model is motivated by the aim of minimizing driving distances. Depending on the time needed to cultivate each single lot the relevant measure could be anything between the (weighted) length of a star rooted at the farmhouse or the length of a minimum traveling salesman tour from lot to lot. For computational simplicity we actually compute for each farmer f a minimum spanning tree in the complete graph G_f in the Euclidean plane whose vertices are the centers of the lots assigned to f . Denoting by $t(f)$ the length of a minimum spanning tree in G_f , we take

$$T = \sum_{f=1}^{f^*} t(f)$$

as a measure for the total length of driving distances.

Recall that in order to relax the problem to linear programming we have to approximate the $f^*(f^* - 1)$ -dimensional unit ball (i.e., dimension 20 in the first example, 156 in the second) by a polytope. Figures 1 up to 3 depict the status quo and solutions obtained by means of random approximation and by a Hadamard-matrix based approximation in the first example.⁵ Here the random polytope P_r is created by choosing uniformly at random 40 points c from the 20-dimensional unit sphere and adding the constraints $c^T x \leq 1$. In the Hadamard-matrix based approximation we use the columns (and their negatives) of a (16×16) - and a (4×4) -Hadamard-matrix to (deterministically) construct a 40-facet approximating polytope.

Figures 4, 5 and 6 depict the status quo and the solutions obtained by means of random approximation and approximation based on Hadamard-matrices for our second sample region, respectively.

⁴ Because of the data protection act we give information only as specific as needed for the purpose of this paper.

⁵ The original multicolor figures can be viewed under <http://www-m9.ma.tum.de/dm/consolidation/OR2003>

The quadruple $(a, b, c/d)$ given with the figures indicates the relative increase (percentage) of the square of the Euclidean norm (a), the relative decrease in T (b), respectively, while c and d are tight lower and upper bounds for the relative loss/gain of farmsize in the worst case.



Fig. 2. Random polytope solution: (1074.4;-42.7;-0.37/0.77)



Fig. 3. Hadamard matrix solution: (850.49;-42.6;-0.66/1.07)



Fig. 4. Status quo in a second region



Fig. 5. Random polytope solution: (1222.0;-57.9;-0.29/2.73)



Fig. 6. Hadamard matrix solution: (1196.0;-58.7;-0.65/7.50)

References

1. A. Brieden, *On geometric optimization problems likely not contained in APX*, Discrete Comp. Geom. 28 (2002), 201–209.
2. A. Brieden, *On the approximability of (discrete) convex maximization and its contribution to the consolidation of farmland*, Habilitationsschrift, Tech. Univ. Munich, Juni 2003.
3. A. Brieden and P. Gritzmann, *On the inapproximability of polynomial programming, the geometry of stable sets, and the power of relaxation*, The Goodman-Pollack Festschrift, Springer, New York (2003), 303–313.
4. A. Brieden and P. Gritzmann, *Optimal k -clustering in Euclidean space*, Preprint.
5. A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits, *Approximation of radii and norm-maxima: Randomization doesn't help*, Proc. 39th IEEE FOCS, 1998, pp. 244–251.
6. A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits, *Deterministic and randomized polynomial-time approximation of radii*, Mathematika 48 (2001), 63–105.
7. A. Brieden, P. Gritzmann, and V. Klee, *Inapproximability of some geometric and quadratic optimization problems*, In: Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems (P.M. Pardalos, ed.), Nonconvex Optimization and its Applications, vol. 42, Kluwer (Boston), 2000, pp. 96–115.
8. H.L. Bodlaender, P. Gritzmann, V. Klee, and J. van Leeuwen, *Computational complexity of norm-maximization*, Combinatorica 10 (1990), no. 2, 203–225.
9. P. Gritzmann and V. Klee, *On the 0-1-maximization of positive definite quadratic forms*, Operations Research Proceedings 1988 (Berlin) (D. Pressmar et al., eds.), Deutsche Gesellschaft für Operations Research, Springer, 1989, pp. 222–227.
10. P. Gritzmann and V. Klee, *Computational complexity of inner and outer j -radii of polytopes in finite dimensional normed spaces*, Math. Programming 59 (1993), 163–213.

A Bipartite Graph Simplex Method

Reinhardt Euler¹

Faculté des Sciences, B.P.809, 20 Avenue Le Gorgeu,
F-29285 Brest Cedex, France

Abstract. We introduce the data structure of a *rooted, alternating, labelled tree* and show its utility for a combinatorial solution of the weighted stable set problem in bipartite graphs via linear programming. As a by-product we obtain a weighted max-min relation. We illustrate the algorithm on grid graphs and present computational results that indicate the efficiency of our method.

1 Introduction

An undirected graph $T=(V,E)$ without cycles is a *forest* and a connected such graph is a *tree*. If one of the vertices, say r , is distinguished, we speak of a *rooted tree*. In a rooted tree there is a unique path $p(x,y)$ between any two vertices x and y , and we are particularly interested in those paths which connect the vertices to the root. The length (in number of edges) of such a path $p(x,r)$ gives us the *distance* of x to r . The vertex set V of a rooted, *alternating* tree partitions into V_w and V_b , the set of (white) vertices at even, and the set of (black) vertices at odd distance from the root. Also observe, that both these vertex sets are *stable*, i.e. no two of their members are joined by an edge. Finally, we let $N(x)$ denote the neighborhood of vertex x within T for any x . A classical problem in combinatorial optimization is the *weighted stable set problem* (P) in a graph $G=(V,E)$: given a weight function $c : V \rightarrow \mathbb{R}$, find a stable set of maximum total weight. In this paper, we are interested in solving this problem for bipartite graphs. The idea is to adapt the simplex method to exploit the special structure of (P) similar to the network simplex method (cf. Dantzig [3], Cunningham [2] or Ahuja, Magnanti, Orlin [1] for a comprehensive presentation). We just recall that the formulation of network flow problems as linear programs involves the vertex-arc incidence matrix of the underlying network, which is well known to be totally unimodular. The formulation of (P) as a linear program makes use of the edge-vertex incidence matrix of the underlying graph. In our case, this matrix is also totally unimodular, and this property even characterizes bipartite graphs. It happens that problem (P) can be converted into a network flow problem, but in view of its structural properties we felt that a more direct method for the solution of (P) could be developed. It turns out that rooted alternating trees become a very useful tool, in particular in their *labelled* form: a rooted, alternating tree $T=(V,E)$ is *labelled* if the vertices are labelled with integers c_v and the edges with integers π_e as obtained by the following procedure:

Edge labelling:

- i) initialize π_{ij} to 0 for all $ij \in E$.
- ii) Starting from any vertex i add and subtract alternately c_i from π_{jk} along the edges jk of the unique path from i to the root.
- iii) Label the root with the value $\tilde{c}_r := c_r - \sum_{j \in N(r)} \pi_{rj}$.

We just mention that rooted trees have applications in various fields, especially in computer science and operations research. We also think that the interest of such trees is not limited to bipartite graphs: since the latter form a special class of perfect graphs, an appropriate generalization of these trees might help to formulate a combinatorial algorithm for solving problem (P) in perfect graphs. Let us explain our approach by a small example. Fig. 1 exhibits a tree, in which the \tilde{c}_r is indicated as the root's second label:

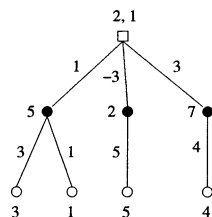


Fig. 1. A rooted, alternating, labelled tree

A closer look at this example shows the following:

- a) $\sum_{j \in N(i)} \pi_{ij} = c_i \quad \forall i \in V \setminus r$.
- b) Some of the π_{ij} are < 0 and $\tilde{c}_r = 1$.
- c) The total weight of V_w is 15 and that of V_b equals 14.

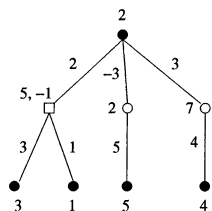


Fig. 2. The same tree with $\tilde{c}_r < 0$ and a negative edge label

Now let us look at Fig. 2, in which we have interchanged the black against the white vertices, chosen a new root among the white ones, recalculated one edge and one root label. V_b is now of total weight 15, but we can still do better, and this is due to a negative edge label: deletion of the corresponding edge and another root change leads us to the forest represented in Fig. 3: we observe that vertex set V_b is now of total weight 17 and it is even of maximum weight: \tilde{c}_{r_1} and \tilde{c}_{r_2} are ≤ 0 and the edge labels are all ≥ 0 , which is nothing but the optimality certificate of the simplex method.

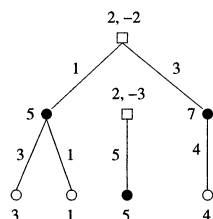


Fig. 3. The optimal forest

This paper is organized as follows: we continue to explain the relationship with linear programming by giving an interpretation of the corresponding simplex tables in purely graph-theoretic terms. In section 3 we present the general algorithm (illustrated on a grid graph in the appendix) together with a general max-min relation that follows in a straightforward way. Section 4 is on computational experience as obtained from an implementation of our algorithm in C (cf. [4]) applied to a series of randomly generated problem instances over grid graphs. We conclude with some remarks on future work.

2 Bipartite Graphs and the Simplex Method

Given a bipartite graph $G=(V,E)$ we can formulate problem (P) as the following linear program

$$(LP) \begin{cases} \text{maximize } c^T x \\ \text{subject to } x_i + x_j \leq 1 \quad \forall ij \in E \\ x_i \geq 0 \quad \forall i \in V, \end{cases}$$

or written in standard form

$$(LPS) \begin{cases} \text{maximize } c^T x \\ \text{subject to } x_i + x_j + y_{ij} = 1 \quad \forall ij \in E \\ x_i \text{ and } y_{ij} \geq 0 \quad \forall i \in V \text{ and } ij \in E. \end{cases}$$

Throughout the following we will use the slack variables as variables on the edges of G . It is well known that any basic feasible solution of (LPS) corresponds to a stable set in G and vice versa, and therefore, (P) can be solved by the simplex method. Also, for an efficient implementation only graph-theoretic concepts should be used: the notion of a rooted, alternating, labelled tree, and more generally, that of a *c-spanning forest* is such a concept:

Definition 1 Given a bipartite graph $G=(V,E)$ together with a weight-function $c : V \rightarrow \mathbb{R}$, a *c-spanning forest* F in G is a collection of (vertex disjoint) rooted, alternating, labelled trees $T_i = (V_i, E_i)$ with $V_i \subseteq V, E_i \subseteq E$ for $i = 1, \dots, m$ such that every vertex v of G is contained in one of the V_i and no two vertices at odd distance from their root are connected in G . F is called feasible, if all edge labels π_{ij} are positive and all root labels \tilde{c}_r non-positive.

We just recall from our introductory example that vertices with non-positive weight can be treated as single roots, and that for the non-roots we have $\sum_{j \in N(i)} \pi_{ij} = c_i$.

Observation 1 *If c is the $|V|$ -vector of all ones, a feasible c -spanning forest coincides with a minimum size vertex cover.*

One can show by induction that the simplex method applied to (LPS) will produce a series of simplex tables that are all of the same general form (as depicted in Fig. 4): the current basis B partitions into a vertex set V_B and an edge set E_B , and the corresponding variables can have values 0 or 1. The non-basic indices (with their variables at value 0) are either roots or edges; the last line contains their reduced costs and negative edge labels, respectively. Also note that the paths starting at a basic vertex or edge are represented as $(+1, -1)$ -alternating sequences of edges, the first being labelled $+1$.

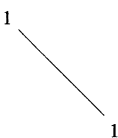
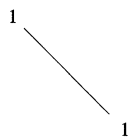
V_B		E_B			
ST:		one non-zero entry per line (+1 if path is odd, -1 if path is even)	0	unique paths from basic vertex to root	0 or 1
	0	two non-zero entries per line (+1 or -1 but at most one -1) (no nonzero entries)		unique paths from basic edge to two roots (cycles)	0 or 1
	0 ——— 0	reduced costs	0 ——— 0	negative edge labels	-z

Fig. 4. A related simplex table ST in general form

Obviously, the simplex table associated with the initial basis of (LPS) is of the required form: it simply reduces to the lower half of table ST. To see how these tables change there are 4 possibilities for a pivot step:

- a root enters the basis and a vertex (resp. edge) leaves it (type 1 resp. 2),
- an edge enters the basis and a vertex (resp. edge) leaves it (type 3 resp. 4).

Examples for these 4 types are the following:

- a root interchange, - connecting 2 trees by an edge to form a single tree,
- deconnecting a tree into 2 new trees and - the modification of edge labels along a cycle.

It is not difficult to show the following

Theorem 1 *The tables encountered during the solution of (LPS) by the simplex method are all of the general form as represented in Fig. 4.*

And since pivoting preserves total unimodularity, we also have

Observation 2 *The coefficient matrix of such a table is totally unimodular.*

It is also straightforward to see how a c-spanning forest can be associated in a unique way with a simplex table ST: for this observe that any basic vertex is associated in a unique way with a root. Together with possibly isolated roots this provides a partition of V into V_1, \dots, V_m . The corresponding trees can now be reconstructed by use of the paths given for any basic vertex: the trees are alternating because a non-basic edge is always incident to a basic vertex at value 1 and to another vertex at value 0. Finally, we can determine the edge labels for any tree by using the $(+1, -1)$ -sequences associated with all paths and the corresponding vertex weights (cf. our introductory example).

3 The Algorithm

The input to our algorithm is a c-spanning forest F that we maintain throughout the execution. Recall that at any stage $\sum_{j \in N(i)} \pi_{ij} = c_i$ for any basic vertex i , which implies that $\sum_{ij \in E} \pi_{ij} = \sum_{i \in V_b} c_i$, the total weight of V_b that we are aiming to maximize. As soon as the reduced costs are non-positive and the edge labels non-negative, by linear programming duality the set V_b is of maximum weight. For combinatorial reasons we may eliminate all edges with 0-label at the end of the procedure, and for notational convenience we call a vertex (edge) whose variable is basic and at value 0 or 1, a *0(or 1)-basic vertex (edge)*.

Bipartite Simplex

Phase 1: Roots

1. If all reduced costs are non-positive, goto Phase 2.
Choose a root r with corresponding tree T_r for which \tilde{c}_r is maximum.
2. Check whether there is a 0-basic edge connecting a vertex at value 0 of T_r to a 1-basic one of another tree $T_{r'}$:
 - 2.1. there is no such edge:
 - if T_r consists of a single root color it black, connect it to a 0-vertex of $T_{r'}$ by a non-basic edge and revise the edge labels on $p(r, r')$ by alternately adding and subtracting \tilde{c}_r ; set $\tilde{c}_{r'} := \tilde{c}_{r'} - \tilde{c}_r$, and make all basic edges incident to r 0-basic.
 - otherwise exchange the black vertices in T_r against the white, choose a neighbor r' of r as the new root, set $\pi_{rr'} := \pi_{rr'} + \tilde{c}_r$, $\tilde{c}_{r'} := -\tilde{c}_r$ and make all 1-basic edges incident (with one vertex) to T_r 0-basic and vice versa. (We have augmented the objective function value z by \tilde{c}_r .)
 - 2.2. there is such an edge:
 - make it non-basic and the root r a 0-basic vertex; revise the edge labels along $p(r, r')$ by alternately adding/subtracting \tilde{c}_r and set $\tilde{c}_{r'} := \tilde{c}_{r'} + \tilde{c}_r$. (T_r and $T_{r'}$ have been connected to form a new rooted, alternating, labelled tree with higher reduced cost.) Goto 1.

Phase 2: Edge labels

1. If all edge labels are positive, STOP.

If there is no edge label with negative value, goto 3.

Choose an edge ij with minimum π_{ij} ; let T be the associated tree rooted at vertex r .

2. Make ij 0-basic to create two subtrees T_i and T_j ; let i be at 0-value.

2.1. r is in T_j :

make i the new root of T_i and set $\tilde{c}_i := \pi_{ij}$. In T_j , revise the edge labels on $p(j,r)$ by alternately adding and subtracting π_{ij} ; set $\tilde{c}_r := \tilde{c}_r - \pi_{ij}$. If $\tilde{c}_r > 0$ goto Phase 1. Else goto 1.

2.2. r is in T_i :

revise the edge labels on $p(i,r)$ by alternately adding and subtracting π_{ij} and set $\tilde{c}_r := \tilde{c}_r + \pi_{ij}$. In T_j choose a neighbor r' of j as the new root, set $\tilde{c}_{r'} := -\pi_{ij}$ and $\pi_{jr'} := \pi_{jr'} + \pi_{ij}$. Goto Phase 1.

3. (Optional) Make ij 0-basic to create two subtrees T_i and T_j ; let i be at 0-value. If r is in T_j , make i the new root of T_i and set $\tilde{c}_i := 0$. If r is in T_i , choose a neighbor r' of j as the new root of T_j and set $\tilde{c}_{r'} := 0$. If all edges with 0-label have been deleted, STOP. Else choose another such edge ij and goto 3.

The validity of the algorithm follows from the fact that the basic operations (steps 2.1, 2.2 of Phase 1, and steps 2, 3 of Phase 2) have an immediate interpretation as pivot steps (of type 1, 2 for steps 2.1, 2.2 of Phase 1, and of type 3 for steps 2, 3 of Phase 2). Moreover, since at every step either the objective function value or one of the reduced costs increases by a positive value, our algorithm is certainly finite. However, we are not able to indicate a firm complexity bound: the deletion of a negative edge label may, for example, engender new negative edge labels. This behavior strongly resembles the network simplex method, cf. [2]; the computational results to be presented in the next section indicate the efficiency of our method in a similar way. As a by-product we obtain the following max-min relation:

Observation 3 *In a bipartite graph the maximum weight of a stable set equals the weight of a feasible c -spanning forest.*

4 Computational Experience

The tests have been run under Linux on a Pentium III 500 MHz with 256 MB of RAM, and our algorithm has been implemented as a 1000 lines C code compiled with the -O4 option of GNU gcc (cf. [4]). The test problems are grid graphs of size pq with vertex weights randomly chosen within the range $[1,100]$. Table 1 reports the running times (in milliseconds) compared with those obtained with the LP-solver *LP-SOLVE* in its version 3.2.

These results clearly indicate the superiority of our algorithm. A comparison with other LP-solvers (such as SOPLEX or CPLEX) is under way: first results indicate a 50 % - superiority of our algorithm over SOPLEX.

Table 1. Bipartite Simplex vs. LP-SOLVE

p	q	vertices	edges	Bipartite Simplex	LP-SOLVE
45	45	2025	3960	486	4736
55	55	3025	5940	1054	11081
65	65	4225	8320	3087	26775
75	70	5250	10355	4121	39538
65	85	5525	10900	6044	46941
80	80	6400	12640	8512	68916
90	90	8100	16020	14256	110052
100	100	10000	19800	23659	177528

5 Conclusions

We already mentioned the interest of a suitable generalization of rooted, alternating, labelled trees for the class of perfect graphs, and the design of a *perfect graph simplex method* along the same lines (involving the associated clique-vertex incidence matrix). Also note that our trees generalize alternating paths, and therefore an interesting relation with matching- (and augmenting path-) theory should exist. Another direction to follow would be to specialize our algorithm to solve problem (P) over planar grid graphs or trees, possibly within firm complexity bounds. One could also think of using these methods to approximate optimal solutions of problem (P) over more general graphs. Finally, the study of different pivot selection rules and other implementational aspects should lead to computational improvements.

References

1. Ahuja, R. K., Magnanti, Th. L., Orlin, J. B. (1993) Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs
2. Cunningham, W. H. (1976) A network simplex method. Math. Progr. 11, 105–116
3. Dantzig, G. B. (1951) Application of the simplex method to a transportation problem. In: Koopmans, T. C.(Ed.): Activity Analysis of Production and Allocation. Wiley, New York, 359-373
4. Lemarchand, L. (2003) An implementation of the bipartite graph simplex method in C. Technical report, University of Brest

6 Appendix

We are now going to illustrate our algorithm on a *grid-graph of size pq*, i.e. a graph whose pq vertices are arranged to form a rectangle of p lines

and q columns, two vertices being connected if they are at euclidean unit distance. These graphs are easily seen to be bipartite; they are planar, too, and therefore our results are also interesting for solving stable set problems in planar graphs. Fig. 5 exhibits such a graph of size $(4,6)$, with integer weights attached to the vertices. At this initial stage, all vertices represent roots, whose weights coincide with the reduced costs, and all edges are 1-basic.

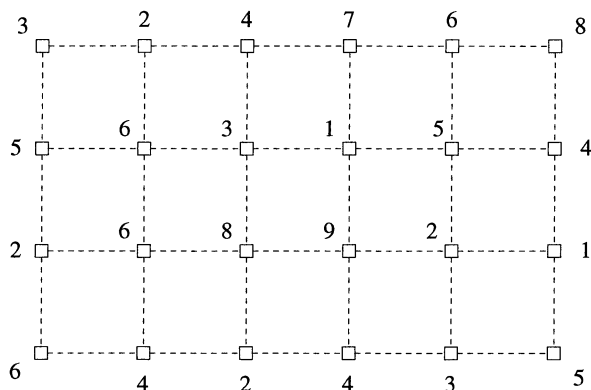


Fig. 5. A weighted grid graph

For convenience we represent the (non-basic) roots by white squares and the 0- and 1-basic vertices as white and black vertices, respectively, to which the weights and reduced costs will be attached. Edges will be black (if non-basic) with their labels attached, or dotted (0- or 1-basic); in Fig. 6 a 1-label is attached whenever such an edge is 1-basic. After having obtained the non-positivity of all reduced costs two negative edge labels remain to be treated (involving step 2.2 of Phase 1). The final, optimal solution is given in Fig. 6.

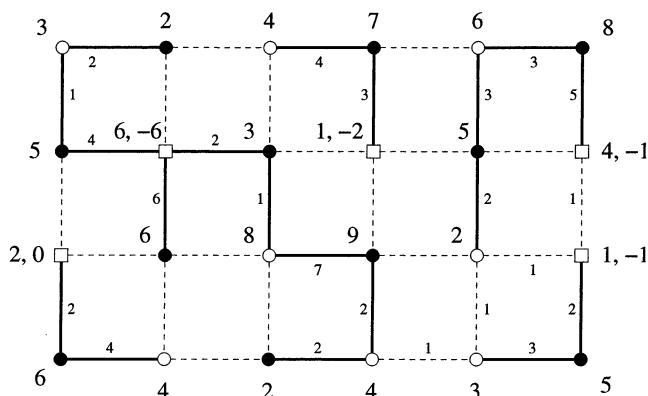


Fig. 6. The optimal solution of total weight 58

Regions of Stability for Nonlinear Discrete Optimization Problems

Diana Fanghänel

TU Bergakademie Freiberg, Germany

Abstract. In this paper we investigate parametric optimization problems of the form $\inf_x \{F(x) - p^\top x : Ax \leq b, x \in \mathbb{Z}^n\}$ with $F : \mathbb{R}^n \rightarrow \mathbb{R}$ being differentiable and convex and $p \in \mathbb{R}^n$ being some parameter. For this problem we consider the regions of stability, i.e. we ask for the set $R(x^0)$ of all parameters p for which a given feasible point x^0 is optimal. Of special interest will be the conditions under which the sets are bounded or polyhedral.

1 Introduction

In the following we want to consider parametric optimization problems of the form

$$\begin{cases} f(x, p) = F(x) - p^\top x \rightarrow \min_x \\ \text{s.t. } Ax \leq b, x \in \mathbb{Z}^n, \end{cases} \quad (1)$$

where $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $p \in \mathbb{R}^n$ is a parameter. The function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is supposed to be convex and differentiable on \mathbb{R}^n .

Let $S_D := \{x \in \mathbb{Z}^n : Ax \leq b\}$ denote the set of all feasible points.

Definition 1. Let $x^0 \in \mathbb{Z}^n$ a feasible point. Then the set

$$R(x^0) = \{p \in \mathbb{R}^n : f(x^0, p) \leq f(x, p) \text{ for all } x \in S_D\}$$

is called region of stability for the point x^0 .

Thus the set $R(x^0)$ denotes the set of all parameters for which the point x^0 is optimal. Let us consider the following example.

Example:

$$\begin{cases} f(x, p) = \frac{1}{2}x^\top x - p^\top x \rightarrow \inf_x \\ -x_1 + x_2 \leq 3 \\ 3x_1 + x_2 \leq 15 \\ -x_2 \leq 3 \\ -3x_1 - 2x_2 \leq 9, x \in \mathbb{Z}^2 \end{cases}$$

We want to describe the regions of stability for all $x \in S_D$. In this example the regions of stability partition the parameter set as shown in figure 1.

The sets $R(x)$ have some interesting properties in this example. For all $x \in S_D$ the regions of stability are convex polyhedra and $R(x)$ is bounded iff $x \in \text{int conv } S_D$.

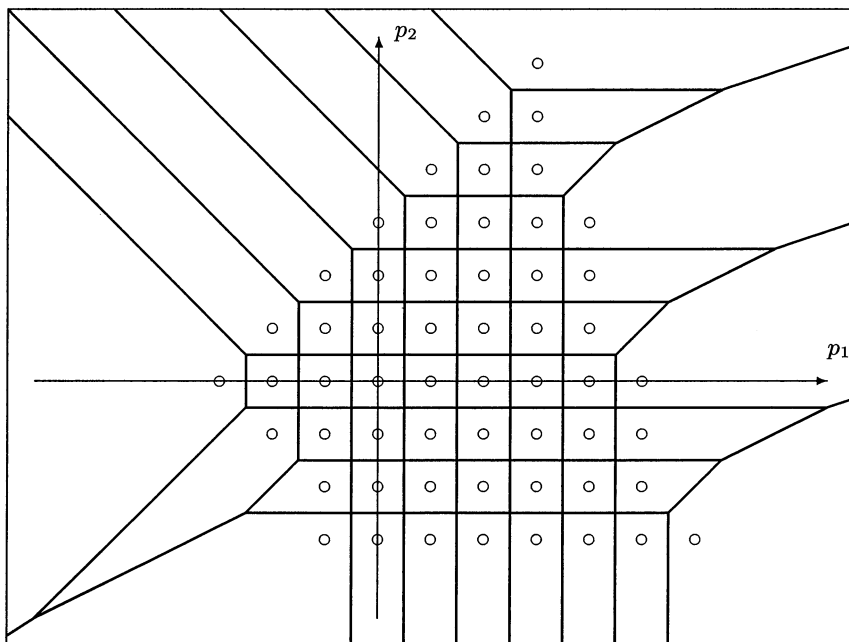


Fig. 1. Regions of stability in the example

2 First Results

Up to now regions of stability have been investigated mostly for integer linear optimization problems ([2],[3]). Further for a more general objective function they were proved to be convex and closed ([4]). We will consider functions of the form $f(x, p) = F(x) - p^\top x$ where $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is supposed to be convex and differentiable. For information about (strict/strong) convexity we refer to [1]. First we want to give some auxiliary results.

Let $x^0 \in S_D$. Then for all $p \in R(x^0)$ it holds

$$\begin{aligned} F(x^0) - p^\top x^0 &\leq F(x) - p^\top x & \forall x \in S_D \\ F(x) &\geq F(x^0) + p^\top (x - x^0) & \forall x \in S_D \\ p^\top (x - x^0) &\leq F(x) - F(x^0) & \forall x \in S_D. \end{aligned} \quad (2)$$

Lemma 1. *Let $x^0 \in S_D$. Then*

1. $R(x^0)$ corresponds to the intersection of (maybe infinitely many) half-spaces.
2. $R(x^0)$ is convex and closed.

This lemma is an easy consequence of the inequalities (2). Now we get the following simple conclusion.

Corollary 1. *Suppose the set S_D has finite cardinality. Then the set $R(x^0)$ is a convex polyhedron for each $x^0 \in S_D$.*

Lemma 2. *Let $x^0 \in S_D$. Then $\nabla F(x^0) \in R(x^0)$, i.e. $R(x^0) \neq \emptyset$.*

This can easily be proved using inequalities (2). Further, for this result the special form of $f(x, p)$ is necessary. If $f(x, p)$ has an other form or if $F(x)$ is not convex then $R(x^0) = \emptyset$ can hold for some $x^0 \in S_D$.

Examples:

1. $\begin{cases} (p^2 + 1)x \rightarrow \min \\ 0 \leq x \leq 1, x \in \mathbb{Z} \end{cases}$ It holds $R(1) = \emptyset$ since $p^2 + 1 > 0$ for all $p \in \mathbb{R}$.
2. $\begin{cases} -x^2 - px \rightarrow \min \\ 0 \leq x \leq 2, x \in \mathbb{Z} \end{cases}$

In this example the function $F(x) = -x^2$ is not convex. The regions of stability are $R(0) = (-\infty, -2]$, $R(1) = \emptyset$ and $R(2) = [-2, \infty)$.

Lemma 3. *Suppose $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is strongly convex and $S_D \neq \emptyset$. Then the following holds:*

1. $\inf_{x \in S_D} f(x, p) > -\infty \quad \forall p \in \mathbb{R}^n$
2. $\forall p \in \mathbb{R}^n \exists x^0 \in S_D$ with $\inf_{x \in S_D} f(x, p) = f(x^0, p)$

From this lemma it follows that for each parameter $p \in \mathbb{R}^n$ there exists some $x^0 \in S_D$ with $p \in R(x^0)$. In the following example we will show that this does not hold in general if F is only strictly convex.

Example: Let be given the problem

$$\begin{cases} f(x, p) = e^x - xp \rightarrow \inf_x \\ x \in \mathbb{Z} \end{cases}$$

Then it holds $\inf_{x \in S_D} f(x, p) = -\infty$ if $p < 0$. For $p = 0$ it holds $\inf_{x \in \mathbb{Z}} f(x, 0) = 0$ but there exists no optimal solution.

3 Boundedness and Dimension of the Regions of Stability

Theorem 1. *For each $x^0 \in S_D$ the set $R(x^0)$ is bounded (and therefore compact) iff $x^0 \in \text{int conv } S_D$.*

Proof. Suppose $R(x^0)$ is unbounded for $x^0 \in S_D$. Since $\nabla F(x^0) \in R(x^0)$ and since $R(x^0)$ is convex and closed there exists some $h \in \mathbb{R}^n$, $h \neq 0$ with $\nabla F(x^0) + \alpha h \in R(x^0)$ for all $\alpha \geq 0$. Thus it holds

$$\begin{aligned} F(x) &\geq F(x^0) + (\nabla F(x^0) + \alpha h)^\top (x - x^0) \quad \forall x \in S_D \\ F(x) &\geq F(x^0) + \nabla F(x^0)^\top (x - x^0) + \alpha h^\top (x - x^0) \quad \forall x \in S_D. \end{aligned}$$

If $h^\top(x - x^0) > 0$ for some $x \in S_D$ then for $\alpha > \frac{F(x) - F(x^0) - \nabla F(x^0)(x - x^0)}{h^\top(x - x^0)}$ the above inequality would not be fulfilled. Thus we obtain $h^\top(x - x^0) \leq 0$ for all $x \in S_D$, i.e. there exists some supporting hyperplane at x^0 to $\text{conv } S_D$. Therefore $x^0 \notin \text{int conv } S_D$.

Suppose $x^0 \notin \text{int conv } S_D$. Then there exists some $h \in \mathbb{R}^n$, $h \neq 0$ with $h^\top(x - x^0) \leq 0 \quad \forall x \in S_D$. Since the function F is convex it holds for all $x \in S_D$

$$\begin{aligned} F(x) &\geq F(x^0) + \nabla F(x^0)(x - x^0) \\ &\geq F(x^0) + \nabla F(x^0)(x - x^0) + \alpha h^\top(x - x^0) \quad \forall \alpha \geq 0 \\ &= F(x^0) + (\nabla F(x^0) + \alpha h)^\top(x - x^0). \end{aligned}$$

Thus, $(\nabla F(x^0) + \alpha h) \in R(x^0) \quad \forall \alpha \geq 0$, i.e. $R(x^0)$ is unbounded.

Theorem 2. Let $F : \mathbb{R}^n \rightarrow \mathbb{R}$ be strictly convex. Then it holds $\nabla F(x^0) \in \text{int } R(x^0)$ for all $x^0 \in S_D$, i.e. all regions of stability have dimension n .

Proof. Let us assume $\nabla F(x^0) \notin \text{int } R(x^0)$. Then there exist sequences $\{p^k\} \rightarrow p^0 = \nabla F(x^0)$ and $\{x^k\} \subseteq S_D$ with $f(x^k, p^k) < f(x^0, p^k)$. Let $0 < \delta < 1 \leq \|x^k - x^0\| \quad \forall k$ and let $z^k = \alpha_k x^k + (1 - \alpha_k)x^0$ with $\alpha_k = \delta / \|x^k - x^0\| \quad \forall k$. Then it holds $\|z^k - x^0\| = \alpha_k \|x^k - x^0\| = \delta$ and $z^k \in S = \{x \in \mathbb{R}^n : Ax \leq b\}$. Thus, $\{z^k\}$ has some limit point $z^* \in S$ with $\|z^* - x^0\| = \delta > 0$. Let w.l.o.g. $\lim_{k \rightarrow \infty} z^k = z^*$. Then it holds

$$f(z^*, p^0) = \lim_{k \rightarrow \infty} f(z^k, p^k) \leq \lim_{k \rightarrow \infty} f(x^0, p^k) = f(x^0, p^0)$$

since

$$\begin{aligned} f(z^k, p^k) &= f(\alpha_k x^k + (1 - \alpha_k)x^0, p^k) < \alpha_k f(x^k, p^k) + (1 - \alpha_k)f(x^0, p^k) \\ &< f(x^0, p^k). \end{aligned}$$

Thus, $F(z^*) \leq F(x^0) + \nabla F(x^0)(z^* - x^0)$ which is a contradiction to F being strictly convex.

4 Statements on Polyhedrality

Theorem 3. Let $R(x^0)$ be bounded for some $x^0 \in S_D$ and suppose $F : \mathbb{R}^n \rightarrow \mathbb{R}$ to be strongly convex. Then the set $R(x^0)$ is polyhedral.

Now we may question whether $R(x^0)$ is polyhedral for all $x^0 \in S_D$. In the following example will show that this does not hold in general. We will need the additional restriction that the matrix A is rational.

Example:

$$\begin{cases} \frac{1}{2}x^\top x - p^\top x \rightarrow \inf_x \\ x_1 \leq \sqrt{2}x_2, \quad x \in \mathbb{Z}^2 \end{cases}$$

Since $\sqrt{2}$ irrational $x = 0$ is the only feasible point, which lies on the line $x_1 = \sqrt{2}x_2$. Therefore the regions of stability for all feasible points unequal zero are polytops. The set $R(0)$ is unbounded but not polyhedral.

Further we will need the following lemma:

Lemma 4. *Let be given some set $S_D = \{x \in \mathbb{Z}^n : Ax \leq b\}$ with $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}_+^m$. Then there exists some finite subset $V = \{v^1, \dots, v^N\} \subseteq S_D \setminus \{0\}$ with $S_D \subseteq \{x = \sum_{i=1}^N \alpha_i v^i : \alpha_i \in \mathbb{Z}_+, i = 1, \dots, N\}$.*

Proof. We define a subset $U \in \mathbb{Z}_+^{2n+m}$ as follows

$$U := \{(b - Ax, x^+, x^-) : x \in S_D \setminus \{0\}\}.$$

Then the Gordan lemma [5] tells us that there exist elements $v^1, \dots, v^N \in S_D \setminus \{0\}$ such that for every element of $x \in S_D \setminus \{0\}$ there exists an index $i \in \{1, \dots, N\}$ with $(b - Ax, x^+, x^-) \geq (b - Av^i, v^{i+}, v^{i-})$. We claim that v^1, \dots, v^N have the required property.

To see this, let $x \in S_D \setminus \{0\}$ and $x \notin \{v^1, \dots, v^N\}$. Then there exists an index $i \in \{1, \dots, N\}$ with $(b - Ax, x^+, x^-) \geq (b - Av^i, v^{i+}, v^{i-})$. It follows that $x - v^i$ is integral and that

$$A(x - v^i) = (b - Av^i) - (b - Ax) \leq 0 \leq b.$$

Hence $x - v^i \in S_D \setminus \{0\}$. Repeating these arguments with $x - v^i$ instead of x and noting that $|x - v^i|_1 < |x|_1$ we obtain after a finite number of repetitions of this process a nonnegative integer representation of x .

Theorem 4. *Suppose the matrix A is rational and $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is strongly convex. Then $R(x^0)$ is polyhedral for all $x^0 \in S_D$.*

Proof. Let be given some $x^0 \in S_D$. Further let be $\varepsilon \in \{-1, 1\}^n$ and

$$\mathbb{S}_\varepsilon := \{x \in S_D : (x_i - x_i^0)\varepsilon_i \geq 0 \quad \forall i = 1, \dots, n\}.$$

Since the matrix A is supposed to be rational the set

$$\mathbb{T}_\varepsilon := \{x - x^0 : x \in \mathbb{Z}^n, A(x - x^0) \leq (b - Ax^0), (x_i - x_i^0)\varepsilon_i \geq 0 \quad \forall i\} = \mathbb{S}_\varepsilon - x^0$$

is a rational polyhedron. Thus from lemma 4 it follows that there exist vectors $v^1, \dots, v^N \in \mathbb{T}_\varepsilon$ with

$$\mathbb{T}_\varepsilon \subseteq \left\{ \sum_{i=1}^N \alpha_i v^i : \alpha_i \in \mathbb{Z}_+, i = 1, \dots, N \right\}.$$

Obviously $v^1 + x^0, \dots, v^N + x^0$ are elements of \mathbb{S}_ε . Thus $R(x^0)$ is a subset of the polyhedron

$$P := \{p \in \mathbb{R}^n : F(v^i + x^0) \geq F(x^0) + p^\top v^i, i = 1, \dots, N\}.$$

Further, some $x \in \mathbb{S}_\varepsilon$ is only needed in the description of $R(x^0)$ if the hyperplane $p^\top(x - x^0) = F(x) - F(x^0)$ is cutting the polyhedron P , i.e. if there exists some $p \in P$ with $F(x) < F(x^0) + p^\top(x - x^0)$.

Suppose $x \in S_D$ is such a point. Then there exist numbers $\alpha_1, \dots, \alpha_N \in \mathbb{Z}_+$ with $x = x^0 + \sum_{i=1}^N \alpha_i v^i$ and the system

$$\begin{cases} v^i{}^\top p \leq F(v^i + x^0) - F(x^0), i = 1, \dots, N \\ (x - x^0)^\top p > F(x) - F(x^0) \end{cases} \quad (3)$$

is solvable. From this follows by a theorem of the alternative that the system

$$\begin{cases} z \in \mathbb{R}_+^N, u \in \mathbb{R}_+^2, u \neq 0 \\ \sum_{i=1}^N z_i v^i - u_1(x - x^0) = 0 \\ \sum_{i=1}^N (F(v^i + x^0) - F(x^0)) z_i - u_1(F(x) - F(x^0)) + u_2 = 0 \end{cases} \quad (4)$$

is not solvable. Thus for $u_1 = 1$ the following system is also not solvable

$$\begin{cases} \sum_{i=1}^N z_i v^i = x - x^0 \\ \sum_{i=1}^N (F(v^i + x^0) - F(x^0)) z_i \leq (F(x) - F(x^0)) \\ z \in \mathbb{R}_+^N. \end{cases}$$

Since $x = x^0 + \sum_{i=1}^N \alpha_i v^i$ we get for $z_i = \alpha_i$, $i = 1, \dots, N$

$$\sum_{i=1}^N \alpha_i (F(v^i + x^0) - F(x^0)) > F(x) - F(x^0).$$

Then we obtain

$$\begin{aligned} \sum_{i=1}^N \alpha_i (F(v^i + x^0) - F(x^0)) &> F(x) - F(x^0) \\ &\geq \nabla F(x^0)(x - x^0) + \theta \|x - x^0\|^2 \\ \sum_{i=1}^N \alpha_i (F(v^i + x^0) - F(x^0) - \nabla F(x^0)v^i) &> \theta \sum_{i=1}^N \alpha_i^2 \|v^i\|^2. \end{aligned}$$

Let be $\nu = \min_{i=1, \dots, N} \|v^i\|^2 > 0$, $\alpha = (\alpha_1, \dots, \alpha_N)^\top$ and $d = (d_1, \dots, d_N)^\top$ with $d_i := F(v^i + x^0) - F(x^0) - \nabla F(x^0)v^i$, $i = 1, \dots, N$. Then it holds

$$\|\alpha\| \|d\| > \theta \nu \|\alpha\|^2, \text{ i.e. } \|\alpha\| < \frac{\|d\|}{\theta \nu}.$$

Thus $\forall x = x^0 + \sum_{i=1}^N \alpha_i v^i$ with $(x - x^0)^\top p > F(x) - F(x^0)$ for some $p \in P$ there is $\|\alpha\| < \frac{\|d\|}{\theta_\nu}$. Therefore there are only finitely many $x \in \mathbb{S}_\varepsilon$ for which system (3) is solvable. Since $S_D = \cup_\varepsilon \mathbb{S}_\varepsilon$ this implies that $R(x^0)$ can be generated by finitely many elements of S_D , i.e. $R(x^0)$ is polyhedral.

References

1. Hiriart-Urruty, C.Lamarechal (1993), *Convex Analysis and Minimization Algorithms I*, Springer-Verlag
2. M.Libura (1976), *Stability regions for optimal solutions of the integer programming problem*, Working paper MPD 3-76, Systems Research Institute, Polish Academy of Sciences
3. J.Seeländer (1980), *Untersuchungen über die Existenz von Gitterpunkten in polyedrischen Bereichen und Anwendungen auf Stabilitätsbetrachtungen*, Dissertation(B), Technische Hochschule Leuna-Merseburg
4. B.Bank et al. (1982), *Non-Linear Parametric Optimization*, Akademie-Verlag, Berlin
5. R.Weismantel (1998), *Test Sets of Integer Programs*, *Mathematical Methods of Operations Research* 47:1-37

Optimization Models for the Containership Stowage Problem

Peer Giemsch¹ and Andreas Jellinghaus²

¹ Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften,
Institut für Anwendungen des Operations Research,
Kaiserstr. 12, D-76128 Karlsruhe, Germany, peer@giemsch.de

² Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften,
Kaiserstr. 12, D-76128 Karlsruhe, Germany, andreas.jellinghaus@inka.de

Abstract. This paper deals with the containership stowage problem. Containers are placed on the ship in a last-in-first-out manner and therefore temporary unloading and reloading in subsequent ports along the route, called shifting, is common and results in high costs. This is true, in particular, if the stowage plan is based only on stability constraints of the ship.

The generating of such schedules depends on the transportation load, the technical constraints and possibilities in the ports, the ship geometry, the sequence of ports visited and some other rather technical constraints (e.g. very heavy containers or hazardous goods).

We will show how this containership stowage problem can be modeled as a mixed integer programming model and discuss the computational complexity of the problem. Based on these results, solution methods are developed and some special cases are analyzed.

Furthermore, as a cross reference, we draw our attention to other combinatorial problems like the three-dimensional packing problem with special precedence constraints or pile-up problems. Finally, we propose possible directions for further research.

1 The Containership Stowage Problem

Containerships call at many ports and at each port containers are loaded and unloaded. The containers on board of a containership are put into stacks. A container is only accessible if it is on the top of the stack (last-in-first-out). The task of determining a good container arrangement is called stowage planning, e.g. [1]. Because modern containerships carry up to 7000 containers and visit up to 20 ports, optimization requires solution of a large scale combinatorial problem. To come up with a good stowage plan there might be multiple objectives that could be pursued. We attempt to minimize the container shifts, but there are other objectives as well like (see [14]):

- minimizing usage of ballast water,
- minimizing torsional and shear forces,
- maximizing utilization of the terminal equipment,

- minimizing trim,
- effective hatch usage.

In addition to the accessibility constraint named above, a huge set of other constraints, such as maintain ship stability, requirements for the storage of hazardous cargo, deck strength limits, electric supply of refrigerated units, and limited mixture of 20' and 40' containers.

Also, there are some more limitations for the container terminal and in general the loading of a ship begins before all containers to be shipped have arrived. This makes the problem stochastic and results in several updates of the stowage plan.

2 Basic Model

In the basic model (as given e. g. in [4]) the containership is modeled as a single bay as an array with R rows (height of stacks) and C columns (total number of stacks). There are N ports. The ship starts with an empty bay at port 1 and at port N all containers will be unloaded. The placement of the containers remains unchanged between port i and port $i + 1$. We call the bay uncapacitated if $R = \infty$, capacitated otherwise.

The transportation load is given as a $(N - 1) \times (N - 1)$ matrix $T = (T_{ij})$, where T_{ij} is the number of containers originating at port i with destination j . There is one standard size of container, which can be placed on arbitrary positions in the shipbay. The transportation load should be feasible in respect of hold and deterministic at every port.

There is one crane at every port which can lift any container on the quay and on the bay without limitations. All containers originating in that ports, respectively, have arrived before the ship. The shifting cost per container is equal to 1.

Definition 1 ([1]). A container u of a column is *overstowed* when it blocks the retrieval of another container v and the destination port j of u is later on the schedule than the destination port i of v .

If there are overstowed containers *shiftings* become necessary, i. e. to temporarily unload the containers and reload them later.

Definition 2 (uCSP). Given a transportation matrix, a bay, and a nonnegative integer s , the *unrestricted containership stowage problem (uCSP)* is the decision problem if there exists a stowage plan that cause at most s shifts. Associated optimization problem is to minimize shifts.

The term "unrestricted" means, that there are no technical restrictions to be considered in a sense that each container can be placed in any empty slot.

Theorem 1 ([4]). *Let C be the number of columns in an uncapacitated bay. Then, the uncapacitated, unrestricted containership stowage problem is \mathcal{NP} -complete for $C \geq 4$.*

There are some special cases. Clearly every bay with $N - 1$ uncapacitated columns can result in a zero-shifts plan for every transportation matrix, but it can further be reduced:

Theorem 2 ([4]). *Let T be a $(N - 1) \times (N - 1)$ transportation matrix with no zero entry on or above the diagonal. Then the minimum number of uncapacitated columns C^* needed for a zero-shifts plan to exist is equal to $\lceil \frac{N}{2} \rceil$.*

The common formulation of the uCSP is (see in [2]):

$$\text{Min} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{r=1}^R \sum_{c=1}^C \sum_{v=i+1}^{j-1} x_{ijv}(r, c)$$

subject to

Shipquant ($i = 1, \dots, N - 1, j = i + 1, \dots, N$):

$$\sum_{r=1}^R \sum_{c=1}^C \sum_{v=i+1}^j x_{ijv}(r, c) - \sum_{k=1}^{i-1} \sum_{r=1}^R \sum_{c=1}^C x_{kji}(r, c) = T_{ij} \quad (1)$$

Slot ($i = 1, \dots, N - 1, r = 1, \dots, R, c = 1, \dots, C$):

$$\sum_{k=1}^i \sum_{j=i+1}^N \sum_{v=i+1}^j x_{kji}(r, c) = y_i(r, c) \quad (2)$$

OnTop ($i = 1, \dots, N - 1, r = 1, \dots, R - 1, c = 1, \dots, C$):

$$y_i(r, c) - y_i(r + 1, c) \geq 0 \quad (3)$$

Shift ($j = 2, \dots, N, r = 1, \dots, R - 1, c = 1, \dots, C$):

$$\sum_{i=1}^{j-1} \sum_{p=j}^N x_{ipj}(r, c) + \sum_{i=1}^{j-1} \sum_{p=j+1}^N \sum_{v=j+1}^p x_{ipv}(r + 1, c) \leq 1 \quad (4)$$

$$x_{ijv}(r, c) \in \{0, 1\} \quad y_i(r, c) \in \{0, 1\}$$

We define $x_{ijv}(r, c) = 1$ if there is a container in slot (r, c) loaded on board in port i with final destination j , and unloaded in port v and 0 otherwise. Furthermore $y_i(r, c) = 1$ if upon sailing from port i , slot (r, c) is occupied by a container and 0 otherwise.

The constraints allow for the following interpretation: (1) specify the number of containers to be shipped. (2) ensure that there is at most one container in each route segment. (3) are needed to ensure that the containers are stored in stacks and (4) define shifting movements. The objective counts the shiftings caused by overstowed containers.

Other formulations of the CSP as a mixed integer program can be found in [5] and in [11], but there are even more variables and constraints in use.

3 Extensions of the Basic Model

There are a lot of potential expansions of the basic model, in particular with regard to technical constraints. For example, heavy containers should be loaded in the columns first. Or the containers which need electricity have to be stowed separately.

Different costs of container movements in ports can be incorporated introducing weights into the objective function.

The provision for the ship geometry can be carried out through the usage of virtual containers to be unloaded in a port $N + 1$ or an appropriate data structure other than a matrix for the bay plan.

Conspicuous it is that stowage plans generated on the basic model tend to group containers with the same destination. This is certainly a plausible way if there is only one crane at a terminal. But it might be unfavorable in cases where more than two cranes are available because the grouped containers are only accessible by one crane.

In [12] an integrated approach which combines stowage planning and selection of loading sequences of the quay cranes is presented. Especially the just-in-time arrival of containers at the terminal is modelled.

In all cases mentioned above the basic model is not suited without arranging columns in two dimensions.

We augment the approach in that we include several additional constraints and we examine how given solution methods and data structures have to be modified.

First: Ships are never completely empty. This makes it usually impossible to arrange all containers in order leaving the first port.

Second: Ports visited are not all different as some ports are visited twice, e. g. a typical tour between Europe and Asia is calling Shanghai and Hong Kong two times.

Third: Ships make round trips. Therefore the transportation matrix cannot be defined as above mentioned as there will be no transports between port i and $i + N$ or other pairs.

4 Solution Methods

Due to the fact CSP is \mathcal{NP} -complete and the formulations as a binary linear program does not provide us with much hope to obtain results in reasonable time developing heuristics seems suggestive. There are some methods for producing solutions like simulation, heuristics, rule-based expert systems and decision support systems. We want to mention here [15] as a good starting point. Wilson et al. in [15] and [14] use a two stage process of computerized planning i. e. generalized placement strategy as well as specialized placement procedures. They incorporate many of the technical constraints in the objective function and their procedure models the human planner's conceptional

approach.

We examined the basic model with our own extensions mentioned above.

The generation of random instances was conducted in the following way:

Departing from the empty transportation matrix, a given number of ports, and the capacity of the ship generation of the transportation matrix is done by adding several transports. Input factors are the number of transports to add, and the maximum size for an additional transportation.

A single transportation is added by selecting a random valid (i, j) combination ($j > i$) and adding containers. The number of containers is set to a random value from one to maximum for an additional transportation. If that amount of containers can be added to the transportation matrix without exceeding the ship capacity, it is done. If not, a new random value is selected up to the amount that can be added without exceeding the ship capacity, then that new amount of containers is added to the transportation matrix.

An alternative generation method tries to approximate real world transportation problem. Suppose that several ports are given, and the route of the containership is given as a string of indices to these ports, let n be the length of the route. The container ship is expected to cruise that route several times, the generation process approximates this loop on the given route m times. The whole journey covers $N = n * m$ ports.

Further, for example, a ship on a route linking Europe and China is not expected to carry many containers for transportation within Europe only or within China only, and is not expected to transport containers from Europe to Europe via China. Such restrictions are implemented with a port to port matrix containing weighting factors for each combination of source and destination port for the given route.

As with the simple generation of the transportation matrix, a number of transports are added to this matrix. A new transportation is selected with a random starting port $i \in \{1, \dots, N - 1\}$ and a random destination port $j \in \{i + 1, \dots, i + n - 1\}$. The port associated with the indices i and j are looked up. If the vessel will reach the destination port after i and before j , then no transportation is added. Otherwise a transportation is added in the same manner as in the above named simple generation, but using the additional weighting factor from the port to port matrix. In case the weighting factor is 0, no transportation is added to the transportation matrix.

The generated instances can be downloaded from the web [8].

The stowage plans for all transports steps are build up in parallel by the following three steps of our heuristic. The first step assigning whole columns (inspired by [2]). All steps start processing transportations with maximum distance first. All transportation amount in the transportation matrix T are divided by the row height R . The numbers of whole columns a transportation can fill are noted for processing in the first step, the remaining transportation amount is left in the transportation matrix for further processing in the second and possibly third step.

At the first step whole columns are filled with containers assigned to the same cell for the whole journey. As columns are filled with containers of the same source and destination port only, no shifting is necessary. This step always succeeds, but can leave some columns fragmented in transportation steps, as the column is filled with containers only for some transportation steps but not for all.

At the second step the heuristic tries to fill the remaining containers into columns without causing shifts. Containers are assigned to partially filled columns containing containers with the same destination and the same or an earlier source port. Thus no shifts are created assigning the containers on top of the already assigned containers. Otherwise containers are assigned to columns empty during the required time, so the container can be assigned without the need for shifts.

Remaining containers are filled in the third step one by one: for every container the best assignment is calculated with a branch-and-bound algorithm to minimize shifts necessary to store that container. This algorithm selects several time slices, where there is at least one column for every time slice with a free cell during the given slice, and the time slices add up to the transportation duration of the container to be assigned. Combinations with smaller intersections of the time slices are preferred.

Containers to be shipped from port i to port $i + 1$ can never cause a shift, if these containers are placed last on top of all other containers and are removed first. The first and second step do not assign such transportations.

We compared its results with the suspensory heuristic which was implemented according to the outline Avriel et al. [3]. We could improve results. Our results are available online at [8].

As placement of a column in ship bay is not specified, it is further possible to try to meet stability constraints via adjustment.

5 Related Problems

In this section we show some other problems connected with the CSP. A discussion of Tower of Hanoi problem (see [9] and for an application [10]) or sorting permutations by stacks (see [7]) are omitted due to the scope of this paper.

5.1 Tram Dispatching

In a recent PhD-Thesis [16] the tram dispatching problem is analyzed. In this problem it has to be decided how incoming trains have to be distributed to tracks in the depot so that no shunting occurs in the next morning. The difference to the CSP is that no trains leave the depot before all trains have arrived and this difference is crucial. The thesis additionally analyses the online case. The online case means just in time changes as crash of trains

and delays and are like result changes in the stowage plan if new containers arrive to load. So this approach can possibly be adapted to the stochastic CSP.

5.2 Packing with discharge conditions

The vehicle routing problem with packing constraints was recently presented [13]. In this problem, there are some vehicles with dynamic pickup and delivery at the customers and the geometric load dimensions are also considered and not only the total weight or volume of boxes to be picked. The goal is to minimize tour length and unutilized volume.

In a new approach we analyze the loading and unloading operations by each delivery. Because the volume of each vehicle is limited and there is normally only one access point at the rear, it may be necessary to unload some boxes temporarily. These shifts depend on the sequence of the visited customer and the packaging pattern.

In general we consider a three-dimensional packing problem according to Dyckhoff's [6] classification as $3/V/O/C$. This denotes a three-dimensional packing problem where one large object has to be packed with all items of congruent figures. If every item has a destination and there is a certain sequence of destinations it is useful to think about the discharging procedures. One can identify two cases:

In the first case all items have to be loaded in the beginning. Then, the objective is to minimize, for example, the unutilized space and to minimize shifting operations at each destination by unloading the items.

The second case is more complicated when loading and unloading occurs at every destination as is the case in vehicle routing problems with packing constraints explained above.

In contrast to CSP, the items are not loaded in stacks and there might be very complicated patterns of packed items. The boxes are overstowed in different manners, e. g. there are overstowed boxes above but also in front or beside a box. Clearly there is only one type of boxes in CSP.

6 Further Research

The complexity of the uCSP is still unknown for fixed number of rows R . Also it can be shown that for a single uncapacitated column there exists a polynomial time algorithm [1], the complexity of the uCSP remains unsolved for $C = 2$ or $C = 3$.

There is no application of these models and solutions methods to real data. Many solution methods seems to be ad hoc so that a comprising analysis is auspicious.

A new formulation as a mixed-integer program which can be expanded to multicriteria objectives would be eligible.

References

1. Aslidis, A. (1990) Minimizing of Overstowage in Container Ship Operations. *Operations Research* 90, 457-471
2. Avriel, M. and Penn, M. (1993) Exact and Approximate Solutions of the Container Ship Stowage Problem. *Computers and Industrial Engineering* 25, 271-274
3. Avriel, M.; Penn, M.; Shpirer, N. and Witteboon, S. (1998) Stowage Planning for Container Ships to Reduce the Number of Shifts. *Annals of Operations Research* 76, 55-71
4. Avriel, M., Penn, M. and Shpirer, N. (2000) Container Ship Stowage Problem Complexity and Connection to the Coloring of Circle Graphs. *Discrete Applied Mathematics* 103, 271-279
5. Botter, R.C. and Brinati, M.A. (1992) Stowage Container Planning: A Model for Getting an Optimal Solution. In: *Proceedings of the IFIP TC5/WG 5.6 Seventh International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design*, VII Rio de Janeiro, Brazil, 10-13 September, 1991, 217-229
6. Dyckhoff, H. (1990) A Typology of Cutting and Packing Problems. *European Journal of Operational Research* 44, 145-159
7. Even, S. and Itai, A. (1971) Queues, Stacks and Graphs. In: Kohavi, Z. et al. (1971) *Theory of machines and computations*. Academic Press, New York London, 71-86
8. Giemsch, P. and Jellinghaus, A.: Test-Instances for the CSP. <http://www.andor.uni-karlsruhe.de/fak/inst/andor/a10-www/csp.html> (last visited 2003-09-05)
9. Hinz, A. M. (1989) The Tower of Hanoi. *Enseignement Mathématique*, II. Séries 35, 289-321
10. Sarkar, U.K. (1998) Solution to Cargo Loading Problem Using Multipeg Towers of Hanoi - A Heuristic Search Approach. In: *Proceedings of the International Conference on Knowledge Based Computer Systems (KBCS-98)*, Mumbai, India, 65-76
11. Schott, R. (1989) *Stauplanung für Containerschiffe*. Vandenhoeck u. Ruprecht, Göttingen (in german)
12. Steenken, D. and Winter, T. and Zimmermann, U.T. (2002) Stowage and Transport Optimization in Ship Planning. In: Grötschel et al. (2002) *Online Optimization of Large Scale Systems*. Springer, Berlin Heidelberg, 731-745
13. Türkay, A. and Emel, E. (2003) Vehicle Routing Problem with Packing Constraints. Presentation on the 5th Euro/Informs Joint International Meeting, July 06-10, Istanbul
14. Wilson, I.D. (1997) The Application of Artificial Intelligence Techniques to the Deep-Sea Container-Ship Cargo Stowage Problem. PhD-Thesis, Conta University of Glamorgan - School of Accounting and Mathematics, Glamorgan
15. Wilson, I.D. and Roach, P.A. (2000) Container Stowage Planning: A Methodology for Generating Computerised Solutions. *Journal of the Operational Research Society*, 51, 1248-1255
16. Winter, T. (1999) Online and Real-Time Dispatching Problems. PhD-Thesis, Technische Universität Braunschweig - Fachbereich Mathematik und Informatik, Braunschweig

Solving the Sequential Ordering Problem with Automatically Generated Lower Bounds

István T. Hernádvölgyi

Department of Information and Communication Technology, University of Trento, Povo, 38100 Trento, Italy*

Abstract. The Sequential Ordering Problem (SOP) is a version of the Asymmetric Traveling Salesman Problem (ATSP) where precedence constraints on the vertices must also be observed. The SOP has many real life applications and it has proved to be a great challenge (there are SOPs with 40-50 vertices which have not been solved optimally yet with significant computational effort). We use novel branch&bound search algorithms with lower bounds obtained from homomorphic abstractions of the original state space. Our method is asymptotically optimal. In one instance, it has proved a solution value to be optimal for an open problem while it also has matched best known solutions quickly for many unsolved problems from the TSPLIB. Our method of deriving lower bounds is general and applies to other variants of constrained ATSPs as well.

1 Introduction

The Sequential Ordering Problem (SOP) is stated as follows. Given a graph G , with n vertices and directed weighted edges with the start and terminal vertices designated. Find a minimal cost Hamiltonian path from the start vertex to the terminal vertex which also observes precedence constraints. An instance of a SOP can be defined by an $n \times n$ cost matrix C , where the entry $C_{i,j}$ is the cost of the edge ij in G , or it is -1 to represent the constraint that vertex j must precede vertex i in the solution path.

The SOP is a model for many real life applications, ranging from helicopter routing between oil rigs [10] to scheduling on-line stacker cranes in an automated warehouse [1].

Most asymptotically optimal solvers model the SOP as an Integer Program. Unfortunately the exact structure of the SOP polytope is not yet fully understood and therefore these methods achieved only limited success. Our approach is state space search. The partial completions of feasible tours form a directed acyclic graph. The lower bounds are derived from abstractions of the original state space and correspond to optimal tour completion costs in the abstract space. The lower bounds are stored in a look-up table which we will refer to as the *pattern database*. They are named so, because the abstraction corresponds to merging states of the original state space according to some syntactic *pattern*. The abstraction mechanism described in this paper is general and it is applicable to other versions of constrained ATSPs as well.

* work done at SITE, University of Ottawa, Canada, istvan@site.uottawa.ca

2 Related Work

Optimal solutions to some instances of the SOP were obtained by Ascheuer *et al.* [2] who used the cutting plane technique. They also employed heuristic tour constructions and improvements to derive actual solutions. Escudero *et al.* [5] used a similar approach but the lower bounds were obtained by Lagrangian relaxation.

The HAS-SOP system of Gambardella *et al.* [6] is a metaheuristic technique. In many instances they obtained the best known upper bounds to unsolved instances. This approach is a form of stochastic search and therefore optimal solutions cannot be guaranteed, however the solutions were obtained very quickly. Similar results using genetic algorithms were achieved by Seo *et al.* [11].

Christofides *et al.* [3] considered state space relaxations first to generate lower bounds for TSPs. They also used a state representation very close to ours. The same approach was also considered by Mingozzi *et al.* [9] for the TSP with time windows and precedence constraints.

The pattern database technique was invented by Culberson and Schaeffer [4] and was later used by Korf [8] to obtain optimal solutions to the Rubik's Cube for the first time.

3 Lower Bounds

In our representation, a SOP state s corresponds to a partial completion of the tour. It records the current last vertex in this partial tour as well as the vertices which have not been reached yet. This is very similar to the state representation of Christofides *et al.* [3]. The SOP state space S is a lattice with the start state at the apex and the goal state at the bottom. The lattice has n levels, each corresponding to adding a new vertex to a partial tour such that it still satisfies the constraints. These levels are manifested by the edge structure of the lattice. Edges in S only exist between adjacent levels.

The abstract state space S' is also a lattice with the same number of levels as S . However, $|S'| < |S|$, so we can enumerate it efficiently. The abstract lattice is obtained by clustering states of S on the same level. Figure 1 illustrates the conceptual relationship between the original and abstract lattices. S is the original lattice and S' is the resulting abstraction. We chose states on the same levels to cluster and identified the cheapest cost edges entering and leaving these clusters (drawn with bold edges). These edges will be retained to connect the clustered states which are now replaced by a single vertex (shown as filled circles) in S' . The cost of the optimal completion of s' in S' is a lower bound on the optimal completion cost of the preimage states of s' in S . This is exactly what our lower bounds correspond to. The abstractions are simple relabeling functions we call *domain abstractions*. Initially we assign a unique label to each vertex in the SOP. 0 and $n - 1$ are the labels of

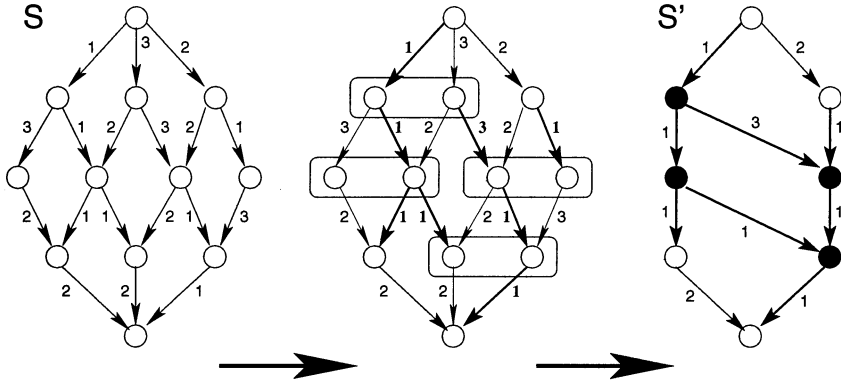


Fig. 1. State Space Lattice Abstraction

the initial and terminal vertices. A domain abstraction is a map from this set of labels to another set of labels of smaller cardinality. We use different domain abstractions at different levels. Let us consider two specific examples of domain abstractions for a 6 vertex SOP.

$$\begin{array}{c}
 v \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\
 \phi_2(v) \quad \hline 0 \quad 1 \quad 1 \quad 2 \quad 1 \quad 3 \\
 \phi_3(v) \quad 0 \quad 1 \quad 2 \quad 2 \quad 3 \quad 4
 \end{array} \quad (1)$$

ϕ_2 and ϕ_3 are applied to levels 2 and 3 respectively. Let \times represent a non-existing edge in S and -1 correspond to a precedence constraint as we described earlier.

$$\begin{array}{c}
 \begin{array}{c} \phi_3 \\ \downarrow \end{array} \begin{array}{c} 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \\ 0 \quad 1 \quad 2 \quad 2 \quad 3 \quad 4 \end{array} \\
 \begin{array}{c} \phi_2 \\ 0 \rightarrow 0 \end{array} \left| \begin{array}{cccccc} \times & 2 & \boxed{1} & \times & 4 & \times \\ \boxed{-1} & \times & \boxed{1} & 4 & \boxed{2} & \times \\ \boxed{-1} & \boxed{1} & \times & 3 & \boxed{3} & \boxed{0} \\ \boxed{-1} & 2 & 1 & 3 & \times & 1 \end{array} \right. \Rightarrow \begin{pmatrix} \times & 2 & 1 & 4 & \times \\ -1 & 1 & 1 & 2 & 0 \\ -1 & -1 & 3 & 2 & 3 \\ -1 & -1 & -1 & -1 & \times \end{pmatrix} \quad (2) \\
 \begin{array}{c} 3 \rightarrow 2 \\ 5 \rightarrow 3 \end{array} \left| \begin{array}{cccccc} -1 & -1 & \boxed{3} & \times & 2 & 3 \\ -1 & -1 & \boxed{-1} & -1 & -1 & \times \end{array} \right.
 \end{array}$$

Equation 2 shows how the original cost matrix is transformed into the cost matrix $C'_{2,3}$ that describes the connectivity and costs between S'_2 and S'_3 . In order to guarantee lower bounds, the most conservative choice is applied to derive the cost representing the merged edges. If all edge costs are -1, then the resulting edge cost is also a -1 (precedences can be preserved). If some are -1

and some are \times we must choose an \times (and eliminate some precedences). If all preimage edges are \times then so is the abstract edge cost (no new edge created). If there are some non -1 and non \times entries then we must choose the one with the minimum cost. In [7], we prove that this construction results in valid lower bounds for S . Although conceptually S' is obtained by compressing S , S is so large that it cannot be enumerated and we build S' by expanding it from the bottom up while also calculating the lower bounds corresponding to the completion costs of the abstract partial tours of S' . For details see [7]. To obtain the abstractions, we minimize a user provided error function of the merged edges. In our experiments, we found that minimizing the absolute error between the merged edge costs and the new edge cost and applying some additional user provided penalty for eliminated precedences is particularly effective. We also used *transposition tables* at levels 3 and 4 to keep track of already visited states and to avoid their re-expansion.

4 Search

Our search algorithms are all derived from depth-first branch&bound. We branch depth-first because memory is reserved to store lower bounds. For the partial tour $s_l \in S_l$, let

$$f(s_l) = c(s_l) + b(s_l) \quad (3)$$

be an estimate of the total cost of the completed tour which is comprised from the cost of the partial tour $c(s_l)$ and its estimated completion cost $b(s_l)$. If $f(s_l)$ exceeds the cost of the best known feasible solution (the incumbent) then it can be pruned. The branches are also sorted in increasing f value order. Plain branch&bound can be greatly improved because of two special properties of the SOP. First, the SOP can be solved by searching from both directions and second, any consecutive sequence of vertices in any feasible tour also corresponds to a smaller SOP by itself. The cost matrix of the subproblem SOP can be obtained by readily taking entries from the original cost matrix and this subproblem can be solved independently. In our experiments we found that search in one direction is often much faster than in the other and therefore it is worthwhile searching from both directions. The switch from one direction to the other is triggered by reaching a user set node expansion limit. The search in both directions works on refining the same incumbent. We named this algorithm bi-directional depth-first branch&bound (or BDFBB). Recursive depth-first branch&bound (or RDFBB) uses a user imposed tolerance limit L . When the number of nodes expanded in the search under a subtree T exceeds L , the subproblem SOP corresponding to the largest subtree not including the start state but including T is solved independently with its own automatically generated lower bounds. We also employed a 3-opt on each new incumbent to opportunistically improve it (also used by [2,6,11]).

Max Width	Random			Absolute Error		
	b_0	ave(b)	Time	b_0	ave(b)	Time
500	400	402.90	0.57	53,965	53,704.10	0.57
2,000	490	830.25	0.80	80,690	59,256.60	0.77
8,000	545	1,219.24	1.00	54,115	52,356.10	1.13
32,000	630	2,146.16	1.46	80,890	65,315.20	1.29
128,000	680	2,934.33	2.13	81,055	60,713.50	2.24
512,000	710	4,925.33	3.30	81,110	69,782.70	2.63
2,048,000	965	7,841.39	5.82	81,350	65,885.20	4.48
8,192,000	1,265	9,704.65	18.43	81,585	70,473.30	12.87

Table 1. Generating Lower Bounds for p43.4

5 Results and Discussion

The two control parameters we supply to calculate the lower bounds are the maximum number of abstract states on a single level (maximum width) and the error function which the abstractions between consecutive levels optimize. As we have mentioned, minimizing absolute error (AE) proved to be effective. This is very clear from Table 1 which shows the average value of lower bounds (ave(b)), the lower bound corresponding to the start state (b_0) and the time it took in seconds to build the pattern databases of various sizes. The problem is one of the previously unsolved ones called p43.4 from the TSPLIB [12]. b_0 is also a lower bound for the SOP itself. The bounds which are tighter than the previously best known value of 69,569 are shown in bold font. We compare randomly chosen abstractions to the ones that minimize AE. The values reported in the columns under “Random” correspond to the best from a pool of 20. Our smallest AE pattern database has much higher lower bounds than the best of 20 random ones which is also 16,384 larger. This experiment also reveals an interesting phenomenon that we encountered on more than one occasions. The AE pattern database with width 2000 is better than many of the larger ones minimizing the same error measure. At this time we have no explanation other than the abstractions seem to preserve the costs between the levels particularly well. With RDFBB, in 22 hours of CPU time we derived that the value of 83,005 is optimal¹. We also have matched the best reported upper bounds within 60 minutes of CPU time to the open problems p43.2, p43.3, p43.4, ry48p.2, ry48p.3, ry48p.4, ft53.2 and ft53.3. In these experiments we used BDFBB and also utilized a 3-opt edge exchange heuristic. We also generated a problem set of 16 SOPs; each over 30 vertices. These are derived from 4 base problems. Two correspond to rounded Euclidean distances on a 500×500 and a 6×6 grid with some random noise

¹ which contradicts a previously reported upper bound of 82,960 whose origin we could not trace

SOP		50,000		250,000		1,250,000	
		1,000 Nodes	Time (sec)	1,000 Nodes	Time (sec)	1,000 Nodes	Time (sec)
500×500	1	3,046,341	21,790.11	20,596	208.29	964	10.57
	2	212,656	1,132.43	8,761	52.63	692	5.27
	3	66,025	278.13	7,750	37.92	460	3.07
	4	27,796	127.25	9,288	49.80	488	3.17
6×6	1	9,022,057	66,451.02	1,775,135	16,058.50	141,987	1,570.72
	2	1,154,906	5,818.16	35,283	206.61	3,476	25.25
	3	26,579	133.14	9,509	57.88	814	5.63
	4	988	4.84	176	1.06	4	0.03
0-1000	1	360,391	2,773.21	34,060	336.05	17,091	184.29
	2	25,661	152.81	3,787	26.34	159	1.51
	3	266,526	1,243.66	70,435	376.39	8,936	60.89
	4	176	1.04	32	0.22	4	0.04
0-10	1	118,071	721.58	33,489	248.34	3,006	32.74
	2	468	3.06	202	1.48	54	0.52
	3	59,310	290.84	22,045	126.68	10,943	61.89
	4	7,788	40.01	895	5.33	43	0.32

Table 2. Nodes Expanded and CPU Times of the Problem Set

added to third of the edge costs. The other two are uniform random costs in $[0, 1000]$ and $[0, 10]$. Next, we generated four random precedence graphs. We applied the constraints implied by these to each of our four base problems and obtained SOPs with 107 (1), 160 (2), 210 (3) and 253 (4) precedence constraints. These include the 57 trivial precedences which are due to the fact that the start and terminal vertices are known. For ease of reference, we named these 16 test problems suggestively. First, we investigate solving these problems with three different size pattern databases of exponentially larger sizes. To compare them we measure nodes expanded in the search and CPU time. For these experiments we did not use the 3-opt tour improvement, only pattern databases. The results are tabulated in Table 2. Our results show that having larger databases pays off. In the case of 500×500-1 allocating a 5 times larger database results in a 148 fold reduction in nodes expanded. Increasing this size yet another 5 times and there is another 21-fold reduction in nodes expanded. While these very high ratios are not typical, our experiments indicate that in general the trade-off between memory and search speed favors adding more memory. It is also the case that, in most cases, the search finishes much faster when the SOP is more constrained. The branch&cut technique of Ascheuer et al. [2] solves the less constrained SOPs faster. The limiting factors in branch&cut, besides the dimensionality, are the size of the pool of inequalities that have to be searched and the construction of the facet inducing inequalities. Non of these are eased by more precedences. For us, more precedence constraints mean less branching in the search space

SOP		50,000					
		Plain		Rec		Rec + Trans	
		K Nodes	Time	K Nodes	Time	K Nodes	Time
500×500	1	3,046,341	21,790.11	405,420	3,220.54	350,431	2,817.33
	2	212,656	1,132.43	351,017	2,113.21	315,387	1,922.89
	3	66,025	278.13	33,327	146.79	26,871	118.18
	4	27,796	127.25	56,153	248.63	44,182	198.21
6×6	1	9,022,057	66,451.02	18,950,825	134,032.34	12,250,761	101,120.71
	2	1,154,906	5,818.16	30,4600	1,750.83	26,8862	1,551.40
	3	26,579	133.14	26,579	135.48	21,293	108.86
	4	988	4.84	988	4.96	988	4.98
0-1000	1	360,391	2,773.21	347,409	2,732.30	317,975	2,506.07
	2	25,661	152.81	17,865	112.49	17,686	109.62
	3	266,526	1,243.66	51,707	230.21	48,021	212.11
	4	176	1.04	176	1.03	176	0.03
0-10	1	118,071	721.58	61,936	420.93	61,798	423.42
	2	468	3.06	468	3.11	468	3.13
	3	59,310	290.84	22,042	111.54	20,171	101.83
	4	7,788	40.01	7,788	40.95	7,786	40.93

Table 3. Nodes Expanded and CPU Times of the Problem Set

and therefore less node expansions. Table 3 compares plain DFBB (Plain) to RDFBB (Rec) and RDFBB with transposition tables (Rec + Trans) with the smallest (maximum width 50,000) pattern databases. The values in bold show improvements that resulted in less search effort than searching with plain DFBB with a 5 times larger pattern database. The improvement is often remarkable but not always present. It requires further investigation to examine the cause as there is a large number of parameters to be set, such as the size of subproblems, databases and expansion limits.

With DFBB, the cost of the initial sequence of feasible solutions is very important since the costs are used as pruning bounds. Interestingly, we found that the size of the pattern database and even the magnitudes of the lower bounds stored have no relevance whatsoever when it comes to obtaining cheap solutions fast. Therefore the use of a 3-opt makes a very big difference. In fact, to this end we believe it would be more effective to start with the cheap and fast solutions of the HAS-SOP system [6] and instead of ordering by f values (Equation 3), we could make use of intermediate solutions of the LP relaxation of the branch&cut solver [2]. The solution to the LP relaxation is a fraction in $[0, 1]$ corresponding to each edge of the SOP. We could interpret these values as probabilities that tell how likely it is that the particular edge is included in the tour. We would sort branches by these probability values but still use our lower bounds for pruning.

We would also like to mention that our lower bounds could be generated when the precedences are arbitrary boolean constraints – such as “vertex

2 or 3 must precede vertex 4” and when some precedence violations are allowed when some user defined penalty is paid. In fact, we believe that pattern databases could be effective to generate lower bounds for a number of constrained Scheduling/Operations Research problems.

I would like to dedicate this work in the memory of my dear friend, Dr. Balázs Zombori, who recently died in a car accident.

This research was supported in part by a research grant provided by the Natural Sciences and Engineering Research Council of Canada. I would also like to thank Dr. Robert C. Holte (University of Alberta) for his insights and encouragement.

References

1. N. Ascheuer. *Hamiltonian path problems in the on-line optimization and scheduling of flexible manufacturing systems*. PhD thesis, Technical University of Berlin, 1995.
2. N. Ascheuer, M. Jünger, and G. Reinelt. A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications*, 17(1):61–84, 2000.
3. N. Christofides, A. Mingozzi, and P. Toth. State space relaxation procedures for the computation of bounds to routing problems. In *Networks*, volume 11, pages 145–164, 1981.
4. J. C. Culberson and J. Schaeffer. Searching with pattern databases. In *Proceedings of the Eleventh Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, volume 1081 of *LNCS*, pages 402–416, 1996.
5. L. F. Escudero, M. Guignard, and K. Malik. A Lagrangian relax-and-cut approach for the sequential ordering problem with precedence constraints. In *Annals of Operations Research*, volume 50, pages 219–237, 1994.
6. L. M. Gambardella and M. Dorigo. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3):237–255, 2000.
7. I. T. Hernádvölgyi. Solving the sequential ordering problem with automatically generated lower bounds. Technical Report TR03-16, University of Alberta, 2003.
8. R. Korf. Finding optimal solutions to Rubik’s cube using pattern databases. In *Proceedings of the Workshop on Computer Games (W31) at IJCAI-97*, pages 21–26, 1997.
9. A. Mingozzi, L. Bianco, and S. Ricciardelli. Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints. *Operations Research*, 45:365–377, 1997.
10. W. Pulleyblank and M. Timlin. Precedence constrained routing and helicopter scheduling: Heuristic design. Technical Report RC17154, IBM, 1991.
11. Dong-Il Seo and Byung-Ro Moon. A hybrid genetic algorithm based on complete graph representation for the sequential ordering problem. In *GECCO-2003*, volume 2723 of *LNCS*, pages 669–680. Springer-Verlag, 2003.
12. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

Scheduling Jobs with a Stepwise Function of Change of Their Values

Adam Janiak¹, Adam Kasperski², and Tomasz Krysiak¹

¹ Institute of Engineering Cybernetics, Wrocław University of Technology,
Janiszewskiego 11/17, 50-372 Wrocław, Poland

² Institute of Industrial Engineering and Management,
Wrocław University of Technology,
Smoluchowskiego 25, 50-372 Wrocław, Poland

Abstract. The paper deals with a single processor scheduling problem in which the sum of values of all the jobs is maximized. The job value is characterized by a stepwise non-increasing function with one or more moments at which a change of job value occur. Establishing an order of processing of datagrams which are sent by router is a practical example of application of this problem. It is proved in the paper, that a special case of the considered problem – with a single common moment of job value change and zero value of jobs after this moment – is NP-hard. Therefore, a pseudo-polynomial time algorithm for the case with common moments of job value change is constructed. It is also constructed and experimentally tested a number of heuristic algorithms which solve the general version of the problem.

1 Introduction

The paper deals with a scheduling problem, where the sum of job values is maximized and a job value is described by a non-increasing stepwise function. Such problem was considered in papers [4] and [5] and in this paper we show some additional results. Besides, the only problem, from among these ones which deal with job values, the problem with an exponential model of job value was investigated ([8] and [1]).

Establishing an order of processing of datagrams which are sent by router is an application example of the problem with a stepwise model of job value. The precise description of this event is given as follows. One of the tasks of router is to determine a path which the datagrams will be sent by farther. Each datagram includes – among other things – a field called *Time To Live* (TTL), which determines how long the datagram could be in a network. A value of TTL is set by sender for each datagram. However, this value is decreased in each router, proportionately to its processing time in the router. But it has to be decreased by at least 1 unit (even if the processing time is shorter than 1 second). If the value of TTL is equal to 0 and the datagram does not achieve the destination, then such datagram is abandoned and the sender receives an error message. The processing of datagram by the router may be called as a job. However decreasing of the

TTL value during the processing datagram by the router may be described by a non-increasing stepwise function. The problem is to find such an order of processing the datagrams in the router that the sum of TTL values of these datagrams (calculated after this processing) is maximal. Beside that, we can meet this problem for example in a process of distributing and selling the commodities with a short sell-by date or in an orcharding (when we establish an order of picking of some kinds of fruits so that the total profit made on selling all fruits is maximal).

The remaining part of the paper is organized as follows. In the next section we give the precise formulation of the problem under consideration and in Section 3 it is proved that its special case – with a single common moment of job value change and the zero value of jobs after this moment – is NP-hard. Section 4 deals with a pseudo-polynomial time algorithm constructed for the problem with common moments of job value change and in Section 5 we present and experimentally compare some approximate algorithms constructed to solve the general version of the problem. Some concluding remarks are given in Section 6.

2 Problem Formulation

There are given a single processor and a set $J = \{1, \dots, n\}$ of n independent and non-preemptive jobs immediately available for processing at time 0. Each job i is characterized by its processing time $p_i > 0$, its value $w_i(C_i)$ calculated at the completion time C_i and the moments $d_{ij} > 0$, $j = 1 \dots k - 1$ at which a change of job value occur. The model of job value is given by non-increasing stepwise function defined as follows

$$w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_{i1} \\ w_{i2}, & d_{i1} < C_i \leq d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < C_i \end{cases},$$

where $w_{i1} > w_{i2} > \dots > w_{ik}$. The objective is to find a schedule π that **maximizes** the sum of job values.

Using the notation $\alpha \mid \beta \mid \gamma$ for scheduling problems [3], the problem considered in this paper is given by

$$1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_{i1} \\ w_{i2}, & d_{i1} < C_i \leq d_{i2} \\ \vdots & \\ w_{ik}, & d_{ik-1} < C_i \end{cases} \right| \sum w_i(C_i).$$

3 Computational Complexity

It is shown now that defined below NP-complete PARTITION problem [2] can be transformed in polynomial time to the decision version of the problem

$$1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_1 \\ 0, & d_1 < C_i \end{cases} \right| \sum w_i(C_i).$$

PARTITION: Given a set $X = \{x_1, x_2, \dots, x_m\}$ of m positive integers for which $\sum_{i=1}^m x_i = 2B$; does there exist a partition of the set X into two disjoint subsets X_1 and X_2 such that $\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i = B$?

Theorem 1. *The problem $1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_1 \\ 0, & d_1 < C_i \end{cases} \right| \sum w_i(C_i)$ is NP-hard.*

Proof. Given an instance of PARTITION, construct the instance of our scheduling problem as follows:

$$n = m; \quad p_i = w_{i1} = x_i; \quad d = y = B; \quad i = 1, \dots, m.$$

It is easy to see that the transformation given above can be done in polynomial time. Now we show that PARTITION has a solution **if and only if** there exists a solution to the constructed instance of our scheduling problem with the value $\sum_{i=1}^n w_i(C_i) \geq y$.

“Only if”. Assume that PARTITION has a solution, i.e., $\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i = B$. Let J_1 and J_2 denote the sets of jobs constructed based on the elements from the subsets X_1 and X_2 , respectively. The processor executes the jobs in the following order: at first the jobs from set J_1 in an arbitrary order and then the jobs from set J_2 in an arbitrary order, as well. The sum of job values obtained for such a schedule is equal to

$$\sum_{i=1}^n w_i(C_i) = \sum_{i \in J_1} w_i(C_i) + \sum_{i \in J_2} w_i(C_i).$$

Since $p_i = x_i$, thus $\sum_{i \in J_1} p_i = B = d$. It means that only the jobs from set J_1 are completed not later than d . Thus we have

$$\sum_{i=1}^n w_i(C_i) = \sum_{i \in J_1} w_{i1} + 0 = \sum_{i \in J_1} w_{i1} = \sum_{i \in X_1} x_i.$$

From the assumption $\sum_{x_i \in X_1} x_i = B$ follows that $\sum_{i=1}^n w_i(C_i) = B = y$, what means that $1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_1 \\ 0, & d_1 < C_i \end{cases} \right| \sum w_i(C_i)$ has a solution.

“If”. Assume now that PARTITION has no solution. It means that $\sum_{x_i \in X_1} x_i \neq B$ and $\sum_{x_i \in X_2} x_i \neq B$ for any subsets X_1 and X_2 . There are two cases

to be considered, namely: a) $\sum_{x_i \in X_1} x_i = B - \lambda$ and $\sum_{x_i \in X_2} x_i = B + \lambda$ and b) $\sum_{x_i \in X_1} x_i = B + \lambda$ and $\sum_{x_i \in X_2} x_i = B - \lambda$, where λ is some positive integer. The criterion value obtained for the case a) is equal to $\sum_{i \in J_1} w_{i1} = \sum_{x_i \in X_1} x_i < B$. For the case b), we have $\sum_{i \in J_1 \setminus \{J_k\}} w_{i1} = \sum_{x_i \in X_1 \setminus \{X_k\}} x_i < B$, where J_k is the set of jobs from J_1 , which complete after d and X_k is the set of elements which correspond to jobs from J_k . \square

4 Pseudo-Polynomial Time Algorithm for the Case with Common Moments of Job Value Change

There exists a pseudo-polynomial time algorithm (PPT) for the problem

$$1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_1 \\ w_{i2}, & d_1 < C_i \leq d_2 \\ w_{i3}, & d_2 < C_i \leq d_3 \\ w_{i4}, & d_3 < C_i \end{cases} \right| \sum w_i(C_i), \text{ which computational complex-}$$

ity is equal to $O(n^2 d_1 d_2 d_3)$. The maximum sum of job values is the value of function $F_j^l(t_1, t_2, t_3)$ where t_1, t_2, t_3 are the state variables. The variable t_i , $i \in \{1, 2, 3\}$, denotes the sum of processing times of such jobs, for which the completion times are larger than d_{i-1} and not larger than d_i , where $d_0 = 0$. The variable l ($l = 1, 2, \dots, n$) denotes the job which is scheduled as the first from among these ones which are completed later than d_1 and not later than d_2 .

The formal description of this algorithm is given below

PPT Algorithm

Step 1: Renumber the jobs according to LPT rule, i.e., $p_1 \geq p_2 \geq \dots \geq p_n$. Set

$$F_0^l(t_1, t_2, t_3) := \begin{cases} w_{l2} & \text{if } t_1 = 0, t_2 = p_l, t_3 = 0 \text{ and } p_l \leq d_2 \\ -\infty & \text{otherwise} \end{cases}. \text{ Then set } j := 1.$$

Step 2: For each $l = 1, 2, \dots, n$; $t_1 = 0, 1, \dots, d_1$; $t_2 = 0, 1, \dots, \min\{d_2 - t_1, d_2 - d_1 + p_l\}$; $t_3 = 0, 1, \dots, d_3 - (t_1 + t_2)$ calculate if $t_2 + t_3 < d_3 - d_2 + p_l$:

$$F_j^l(t_1, t_2, t_3) = \max \begin{cases} F_{j-1}^l(t_1 - p_j, t_2, t_3) + w_{j1}, \\ \quad \text{if } t_1 \geq p_j \text{ and } j \neq l \\ F_{j-1}^l(t_1, t_2 - p_j, t_3) + w_{j2}, \\ \quad \text{if } t_2 \geq p_j \text{ and } j \neq l \\ F_{j-1}^l(t_1, t_2, t_3 - p_j) + w_{j3}, \\ \quad \text{if } t_3 \geq p_j \text{ and } t_3 < (d_3 - d_2) + p_j \text{ and } j \neq l \\ F_{j-1}^l(t_1, t_2, t_3) + w_{j4}, \text{ if } j = l \end{cases}$$

If $j = n$, then go to Step 3, otherwise set $j := j + 1$ and repeat Step 2.

Step 3: Find the optimal schedule by backtracking.

We calculate now the computational complexity of this algorithm. Since we have n different values of j and l , and at most d_i different values of t_i , $i \in \{1, 2, 3\}$, then Step 1 and Step 2 require $O(n^2 d_1 d_2 d_3)$ operations. While Step 3 requires n operations. Thus, the whole PPT Algorithm requires $O(n^2 d_1 d_2 d_3)$ time.

5 Approximate Algorithms for the General Case of the Problem

Since the problem $1 \left| w_i(C_i) = \begin{cases} w_{i1}, & 0 < C_i \leq d_{i1} \\ w_{i2}, & d_{i1} < C_i \leq d_{i2} \\ \vdots \\ w_{ik}, & d_{ik-1} < C_i \end{cases} \right| \sum w_i(C_i)$ is NP-hard,

it is highly unlikely to construct the fast algorithms (i.e. polynomial) which give the optimal solutions of the problem. Therefore, in order to solve it, we construct and experimentally test some approximate algorithms. We present in this section the most interesting results of this research. Thus we describe four fast heuristic algorithms which computational complexity is $O(n \log n)$ and a modification of one of them which computational complexity is equal to $O(kn \log n)$, where k denotes a quantity of moments of job value change. Then we show the results of numerical experiment which compares the efficiency of considered algorithms.

The algorithms $p_i \nearrow$, $w_{i1} \searrow$ and $p_i/w_{i1} \nearrow$ construct the solutions by sequencing the jobs in non-decreasing order of p_i , in non-increasing order of w_{i1} and in non-decreasing order of p_i/w_{i1} , respectively. Two next algorithms M_{mdf} and kM_{mdf} are the modifications of the Moore algorithm ([6], [7]). The M_{mdf} is described below:

Algorithm 1 *The M_{mdf} algorithm*

- Step 1: Set $\pi^1 := \emptyset$, $\pi^2 := \emptyset$ and $J := \{1, 2, \dots, n\}$.
- Step 2: Find a job i^* that satisfies $d_{i^*1} = \min_{i \in J}(d_{i1})$. Put job i^* on the last position in π^1 . Set $J := J \setminus \{i^*\}$ and go to Step 3.
- Step 3: If $\sum_{i \in \pi^1} p_i \leq d_{i^*1}$ then go to Step 4, otherwise find job $k^* \in \pi^1$ that satisfies $p_{k^*}/w_{k^*1} = \max_{i \in \pi^1}(p_i/w_{i1})$ and put it on the last position in π^2 . Set $\pi^1 := \pi^1 \setminus \{k^*\}$.
- Step 4: If $J = \emptyset$ then STOP – the permutation $\pi := \pi^1 \pi^2$ is the expected solution, otherwise go to Step 2.

The kM_{mdf} algorithm is the following modification of the M_{mdf} .

Algorithm 2 *The kM_{mdf} algorithm*

- Step 1: Set $\pi := \emptyset$, $\pi' := \emptyset$, $J^c := \{1, 2, \dots, n\}$, $J^d := \emptyset$ and $j := 1$.
- Step 2: Find a job i^* that satisfies $d_{i^*j} = \min_{i \in J^c}(d_{ij})$. Put job i^* on the last position in π' . Set $J^c := J^c \setminus \{i^*\}$ and go to Step 3.

- Step 3: If $\sum_{i \in \pi} p_i + \sum_{i \in \pi'} p_i \leq d_{i^*j}$ then go to Step 4, otherwise find job $k^* \in \pi'$ that satisfies $p_{k^*}/w_{k^*j} = \max_{i \in \pi'} (p_i/w_{ij})$ and add it to the J^d . Set $\pi' := \pi' \setminus \{k^*\}$.
- Step 4: If $J^c = \emptyset$ then set $\pi := \pi\pi'$, $\pi' := \emptyset$, $J^c := J^d$, $J^d := \emptyset$ and go to Step 5, otherwise go to Step 2.
- Step 5: If $J^c = \emptyset$ or $j = k$ then STOP – π is the solution, otherwise set $j := j + 1$ and go to Step 2.

In the numerical experiment, made on a computer with Pentium II 300 MHz, 176 MB RAM and Windows XP, the parameters values of the problem were randomly generated according to the uniform distribution, and the algorithms were tested for the following different sets of the parameters values:

Set 1: $p_i \in (0, 90)$, $w_{i1} \in [1, 10)$, $k \in [1, 10)$, $d_{ij} \in [d_{ij-1} + 100, d_{ij-1} + 200)$;

Set 2: $p_i \in (0, 90)$, $w_{i1} \in [1, 100)$, $k \in [1, 10)$, $d_{ij} \in [d_{ij-1} + 100, d_{ij-1} + 200)$;

Set 3: $p_i \in (0, 10)$, $w_{i1} \in [1, 100)$, $k \in [1, 10)$, $d_{ij} \in [d_{ij-1} + 5, d_{ij-1} + 15)$;

for $i = 1, \dots, n$ and $j = 1, \dots, k - 1$. Moreover, the values of w_{ij} for $j = 2, \dots, k$ are randomly generated from the following intervals: $w_{ij} \in (0, w_{ij-1})$.

For each set of the parameters values there were randomly generated 500 instances of the problem for each $n = 9, 50, 100, 250$, and 500 jobs. Thus, it was generated 7500 instances of the problem in all. For $n = 9$, for each generated instance the optimal solution was found by explicit enumeration (let OPT denotes the criterion value for this solution). Then, for each algorithm we calculated a performance ratio: $(OPT/ALG - 1) \cdot 100\%$, where ALG denotes an objective function value obtained by the considered algorithm. For $n > 9$, for each generated instance of the considered problem the algorithm which gave a solution with the largest criterion value was found. Let A_{BEST} denote the largest criterion value. Then, the following performance ratio $(A_{BEST}/ALG - 1) \cdot 100\%$ was calculated for the other algorithms. The average values of the performances defined above, obtained for all 500 generated instances of the problem, with a given n and given set of parameters values, are shown in the Table 1, and an average execution times (ms) of the algorithms for Set 1 is given in the Table 2.

As you can see in the Table 1, the values of the problem parameters have an influence on the algorithms accuracy – if $w_{i1} \in [1, 100)$ or $p_i < d_{ij-1} - d_{ij}$ does not hold for each i and j ($i = 1, \dots, n$, $j = 1, \dots, k - 1$, $d_{i0} = 0$) then the accuracy of all considered algorithms is worse. Notice also that the kM_{mdf} algorithm is the most accurate (e.g., for $n > 9$, for most cases it gives the best solutions) but two others ($p_i/w_{i1} \nearrow$ and M_{mdf}) are also accurate – for $n = 9$ for some cases they deliver better solutions than the kM_{mdf} . Moreover these two algorithms are more efficient than the first one (particularly for the large values of k) because of the computational complexity and what follows, because of the execution time. The simple heuristics ($p_i \nearrow$ and $w_{i1} \searrow$) are the least accurate from among the considered algorithms – e.g., they deliver solutions about 10 to 40 percent worse than the optimal ones for $n = 9$.

Table 1. Average performance ratios for the considered algorithms

n	$p_i \nearrow$	$w_{i1} \searrow$	$p_i/w_{i1} \nearrow$	M_{mdf}	kM_{mdf}
Set 1: $p_i \in (0, 90)$, $w_{i1} \in [1, 10)$, $k \in [1, 10)$, $\Delta d_{ij} \in [100, 200)$					
9	5.99	11.13	5.62	4.35	4.07
50	23.88	88.42	13.76	14.37	0.22
100	10.19	84.18	4.48	11.66	0.00
250	7.82	64.19	5.45	6.60	0.28
500	23.16	405.05	12.22	23.87	0.00
Set 2: $p_i \in (0, 90)$, $w_{i1} \in [1, 100)$, $k \in [1, 10)$, $\Delta d_{ij} \in [100, 200)$					
9	8.60	11.92	8.00	5.12	5.26
50	19.73	81.38	1.79	11.37	0.00
100	33.86	145.04	14.93	24.10	0.00
250	22.06	129.09	8.28	18.45	0.00
500	20.11	199.53	9.14	16.90	0.00
Set 3: $p_i \in (0, 10)$, $w_{i1} \in [1, 100)$, $k \in [1, 10)$, $\Delta d_{ij} \in [5, 15)$					
9	17.43	41.67	8.23	16.96	9.35
50	8.59	93.43	8.73	1.23	0.00
100	20.42	186.31	14.96	13.03	0.00
250	20.95	456.22	11.19	13.24	0.00
500	19.65	573.08	7.20	10.51	0.00

Table 2. Average execution times (ms) of the considered algorithms (for Set 1)

n	$p_i \nearrow$	$w_{i1} \searrow$	$p_i/w_{i1} \nearrow$	M_{mdf}	kM_{mdf}
9	0.000	0.000	0.000	0.000	0.000
50	0.000	0.000	0.000	0.000	0.625
100	0.000	1.250	2.813	0.630	1.543
250	6.449	2.832	7.494	7.168	17.702
500	10.000	10.031	30.000	22.622	35.841

To sum up, we constructed three efficient algorithms which solve the general version of the considered problem.

6 Conclusions

In this paper, we considered a problem of scheduling jobs on a single processor in order to maximize the sum of job values. The job value is characterized by a stepwise non-increasing function with one or more moments at which a change of job value occur. It is proved, that a special case of the problem – with a single common moment of job value change and the zero value of jobs after this moment – is NP-hard. Thus, it is highly unlikely to construct an optimal algorithm solving the considered problem in polynomial time. Hence

we constructed a pseudo-polynomial time algorithm for the problem with common moments of job value change and also a number of approximate algorithms for the general version of the problem. The computational complexity of the pseudo-polynomial time algorithm is $O(n^2 d_1 d_2 d_3)$ (where d_i , $i \in \{1, 2, 3\}$, denotes the moments of job value change) thus it is especially efficient for the problems with small values of d_i . For the general version of the problem we constructed two algorithms with computational complexity equal to $O(n \log n)$, which derive the solutions about 10 percent worse than the optimal ones, and one algorithm which is not polynomial one (with computational complexity equal to $O(k \cdot n \log n)$) but it gives better solutions than the two previous algorithms and it can be efficient for problems with small quantity of moments of job value change (the parameter k).

References

1. Bachman A., Janiak A., Krysiak T., Pappis C. P., Voutsinas T. G. (2003) Single machine scheduling problem with job values dependent on their completion times. Sent to International Journal of Production Economics for possible publication
2. Garey M. R., Johnson D. S. (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco
3. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G. (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326
4. Janiak A., Krysiak T. (2003) A single processor scheduling problem with a single change of job value. Accepted for publication in 9th IEEE International Conference Methods and Models in Automation and Robotics
5. Janiak A., Krysiak T., Winczaszek M. (2003) A single processor scheduling problem with a step function of change of job value. Accepted for publication in 9th IEEE International Conference Methods and Models in Automation and Robotics
6. Moore J. M. (1968) An n jobs, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15:102–109
7. Pinedo M. (1995) *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, NJ
8. Voutsinas T. G., Pappis C. P. (2002) Scheduling jobs with values exponentially deteriorating over time. *International Journal of Production Economics*, 79:163–169

Small Instance Relaxations for the Traveling Salesman Problem

Gerhard Reinelt and Klaus M. Wenger

Institute of Computer Science, University of Heidelberg
Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany
{Gerhard.Reinelt,Klaus.Wenger}@Informatik.Uni-Heidelberg.De

Abstract. We explore Small Instance Relaxations in branch-and-cut for the TSP. For small TSP instances of up to 10 cities all facet-defining inequalities of the associated polytopes are known. To exploit this pool of inequalities, we shrink a given TSP support graph to a small graph with at most 10 vertices, search for a violated inequality in the pool, and eventually lift it to obtain a cutting-plane. A Small Instance Relaxation (SIR) is an LP relaxation strengthened by such cutting-planes. We mainly shrink k -way cuts ($k \leq 10$) of weight $k\lambda/2$ to obtain promising small graphs (λ is the mincut weight). For the separation in the low-dimensional space we solve a series of QAP instances. Padberg-Rinaldi shrinking criteria, graph isomorphism detection and facet class selection are applied to avoid unnecessary QAP computations. Our computational results show the usefulness of SIRs for the TSP. We compare SIRs with local cuts by Applegate et al.

1 Introduction

In the TSP we search for a cheapest Hamiltonian cycle or *tour* in the undirected complete graph $K_n = (V_n, E_n)$ with $V_n = \{1, \dots, n\}$ and $n(n-1)/2$ edges $E_n = \{\{1, 2\}, \{1, 3\}, \dots, \{n-1, n\}\}$ weighted by the travel *costs* c_e for $e \in E_n$. The cost of a tour is the total cost of its edges. Let $x = (x_e)_{e \in E_n}$ be a variable vector. Let $W \subset V_n$ and $F \subset E_n$. We define $\delta(W) := \{e \in E_n \mid |W \cap e| = 1\}$ and $x(F) := \sum_{e \in F} x_e$. The Dantzig–Fulkerson–Johnson TSP model is:

$$\begin{aligned} \min c^T x \\ x(\delta(\{v\})) &= 2 && \text{for all } v \in V_n && (1) \\ x(\delta(W)) &\geq 2 && \text{for all } W \subset V_n \text{ with } 3 \leq |W| \leq \lfloor n/2 \rfloor && (2) \\ x_e &\in \{0, 1\} && \text{for all } e \in E_n && (3) \end{aligned}$$

Equations (1) are *degree equations* and (2) are *subtour elimination constraints* (SECs). The feasible solutions of (1)–(3) are called *incidence vectors* of tours. The incidence vector of a tour T is denoted by χ^T . We have $\chi_e^T = 1$ if $e \in T$ and $\chi_e^T = 0$ otherwise. The *Symmetric Traveling Salesman Polytope* STSP(n) for n cities is the convex hull of the χ^T for all tours T in K_n .

The most successful method for solving TSP instances to optimality is branch-and-cut, i.e. branch-and-bound with the cutting-plane method in the

bounding part to strengthen LP relaxations. We assume familiarity with the branch-and-cut idea and the basics of polyhedral theory. We suggest and investigate a way to exploit the facets of $\text{STSP}(k)$ ($k \leq 10$) for cutting-plane generation. See [6,7] for early experiments in this direction.

Let x^* be a fractional solution of an LP relaxation of (1)–(3). Some variables may be fixed to 0 or 1 (branching); the others are relaxed to $[0, 1]$. The relaxation may have been strengthened by cuts. If x^* satisfies (1) and all SECs, we call $(V_n, E_n^* = \{e \in E \mid x_e^* > 0\}, x^*)$ the *TSP support graph* of x^* .

Where do cuts come from? There are general-purpose cuts which do not exploit the problem structure (e.g. Gomory cuts). Further, there are classes of problem-specific cuts (e.g. combs). Considerable effort has been invested in the identification and investigation of such classes. Separation algorithms for them often lag far behind such polyhedral investigations.

We set out to tap a previously almost untapped yet known and available source [4] of cuts for the TSP. We search for cuts in the class consisting of the inequalities obtained by lifting to $\mathbb{R}^{n(n-1)/2}$ the linear descriptions of the small polytopes $\text{STSP}(k)$ for $k = 6 \dots 10$ (Sect. 2). Therefore, a relaxation obtained by adding cuts from this class is called a *Small Instance Relaxation* (SIR). The integrative SIR approach subsumes a wealth of cuts computed by existing separation algorithms for a variety of classical inequality classes. Further, large proportions of the linear descriptions of $\text{STSP}(9)$ and $\text{STSP}(10)$ do not intersect with classes described in the literature. We use a generic separation algorithm as opposed to algorithms tailored to specific inequalities.

The separation algorithm is: For $6 \leq k \leq 10$, we shrink k -way cuts in a given TSP support graph. The shrinking defines a linear map from $\mathbb{R}^{n(n-1)/2}$ to $\mathbb{R}^{k(k-1)/2}$. If the weight vector of a shrunk graph is inside $\text{STSP}(k)$, we fail. Otherwise, we usually get many violated inequalities from the linear description of $\text{STSP}(k)$. We lift these to $\mathbb{R}^{n(n-1)/2}$ using zero-lifting [10] and obtain facet-defining inequalities for $\text{STSP}(n)$ violated by x^* . Many different k -way cuts are chosen (Sect. 4). Generic separation in the low-dimensional space (Sect. 3) is done by solving small instances of the Quadratic Assignment Problem (QAP). This is the bottleneck. We suggest strategies to reduce the number of QAP instances to be solved (Sect. 5).

Applegate et al. [1] suggest an approach which does not care about classes. In a support graph they shrink k -way cuts with k at most about 30. They try to compute a cut in the small space by applying techniques from linear programming and polyhedral theory without running separation algorithms for specific inequality classes. The k -way cuts they choose contain most of V_n in a single subset. They zoom in on small regions. This motivated them to call the resulting cuts, embedded in the Concorde code, *local cuts* [1].

Our SIR approach and the local cut approach have similarities. Both reduce the problem size by shrinking k -way cuts (k small) in support graphs. However, the choice of k -way cuts and the separation in the low-dimensional space differ. Both approaches have interesting aspects (Sect. 7).

2 Low-Dimensional TSP Polytopes

A minimal linear description of $\text{STSP}(k)$ consists of k degree equations and exactly one facet-defining inequality for every facet of $\text{STSP}(k)$. The k independent equations reduce the dimension of $\text{STSP}(k)$ to $(k(k-1)/2) - k$. The last column in Table 1 lists the number of facets at an extreme point. The linear description of $\text{STSP}(10)$ is conjectured to be complete.

Table 1. Low-dimensional TSP polytopes [2,5,7,8]

k	dim	#tours	#facets	#classes	degeneracy
6	9	60	100	4	27
7	14	360	3,437	6	196
8	20	2,520	194,187	24	2,600
9	27	20,160	42,104,442	192	88,911
10	35	181,440	$\geq 51,043,900,866$	$\geq 15,379$	$\geq 13,607,980$

All facets of $\text{STSP}(k)$ for $6 \leq k \leq 10$ are publicly available [4]. For $3 \leq k \leq 5$ the inequalities $x_e \geq 0$ and the SECs defined on two vertices together with the degree equations are sufficient. We do SEC separation first.

The facets in [4] are divided into pairwise disjoint symmetry classes. For each class a representative $f^T y \geq f_0$ is provided. The represented class is

$$\left\{ \sum_{e \in E_k} f_{\pi(e)} y_e \geq f_0 \mid \pi \in \Pi_k \right\} \quad (4)$$

where Π_k are all permutations of $\{1, \dots, k\}$ and $\pi(e) = \pi(\{i, j\}) := \{\pi(i), \pi(j)\}$.

3 Separation in the Low-Dimensional Space

Let y^* be the weight vector of a k -vertex graph obtained by shrinking a k -way cut ($6 \leq k \leq 10$) in (V_n, E_n^*, x^*) . To separate y^* from $\text{STSP}(k)$ we solve a series of QAP instances: one per symmetry class. Brute-force checking of all facets is clearly too slow. Given square matrices (a_{ij}) and (b_{ij}) the QAP is

$$\min_{\pi} \sum_i \sum_j a_{\pi(i)\pi(j)} b_{ij}$$

where π is a permutation. We have symmetric matrices with $a_{ij} = f_{\{i,j\}}$ and $b_{ij} = y_{\{i,j\}}^*$ for $i \neq j$. The main diagonals are 0. In essence, we consider

$$\min_{\pi \in \Pi_k} \sum_{e \in E_k} f_{\pi(e)} y_e^* . \quad (5)$$

Every π for which the sum in (5) is smaller than f_0 defines a cut. This generic separation method can exploit partial linear descriptions. Since the QAP is hard and the number of classes is rising sharply, a small k is desirable.

For exact separation, we use a branch-and-bound algorithm [3] modified so that all π giving a left-hand side smaller than f_0 in (5) are computed. Our QAP instances are sparse. For heuristic separation, we use GRASP-sparse [11]. Per iteration, a good permutation is constructed and locally improved. We can trade off quality against speed by choosing the number of iterations.

4 Generation of Small Graphs by Shrinking

Which k -way cuts should be shrunk and how can they be computed?

Our shrinking is guided by x^* . The representatives $f^T y \geq f_0$ (and the inequalities in (4)) are in TT form [10]. In particular, we have $f_e \geq 0$. Therefore, shrunk graphs with low total edge weight $\sum y_e^*$ tend to keep the left-hand sides in (4) low on average for all symmetry classes. We go for low $\sum y_e^*$.

4.1 Enumeration of Mincut k -Partitions

It is well-known that computing a minimum k -way cut is polynomial in n for fixed k . The best we can get is a partition of V_n into k global minimum weight cuts. We compute all such mincut k -partitions of a TSP support graph (shrunk 1-paths and every vertex is a mincut). While not every graph has such a k -way cut, TSP support graphs usually have many. The literature on minimum k -way cuts mainly deals with $k \leq 6$. We are interested in and our enumeration algorithm works well for $6 \leq k \leq 10$.

We use the cactus representation of all mincuts of a graph [13]. It is as a data structure storing all (that is $\mathcal{O}(n^2)$) mincuts of an n -vertex graph with positive edge weights. The space requirement is $\mathcal{O}(n)$. The inclusion- and intersection-structure of the set of mincuts is captured by the representation. See [13] for a very fast cactus construction algorithm for TSP support graphs. An explicit list of all mincuts takes $\mathcal{O}(n^3)$ space: too much in most cases.

Our algorithm enumerating the mincut k -partitions is polynomial in n for fixed k and requires $\mathcal{O}(n^2)$ space to run. It is an exhaustive enumeration. From experience we know that the enumeration times explode if we get the enumeration strategy wrong. Having chosen some pairwise disjoint mincuts, we try to choose a further mincut not overlapping with the ones already chosen. A key to make the enumeration fast in practice is to consider the mincuts sorted such that their cardinality is non-increasing. This gives effective bounding criteria to keep the enumeration tree small and it allows a fast cactus-based test for overlap. We use a list of ‘mincuts’ consisting of pointers into the cactus data structure which holds the actual mincuts. Intuitively, choosing large mincuts early ‘blocks’ large parts of V_n and subsequent overlap tests get more efficient. For the sake of brevity, we cannot go into details.

For example, on a 2800 MHz PC the algorithm took 2.1 seconds to enumerate all 2,632 mincut 10-partitions of a pr2392 TSP support graph (shrunk 1-paths) having 3,442 mincuts. The cactus construction took 0.03 seconds.

4.2 Randomized Shrinking and Vertex Neighborhoods

Our fully-fledged SIR code uses further methods to compute small graphs.

We randomly shrink edges until we have k -vertex graphs [9]. We choose an edge with probability super-proportional (in [9] proportional) to its weight. This yields (compared to [9]) lower total edge weights $\sum y_e^*$ in the small graphs. Put differently, we compute many near-minimum k -way cuts [9].

We choose vertex neighborhoods $N(v)$ with $|N(v)| = k - 1$. An $N(v)$ together with $V_n \setminus N(v)$ forms a k -way cut. To get shrunk graphs with low total edge weight, we grow neighborhoods by adding vertices that see it by the most weight (max-back order). We also choose neighborhoods by breadth-first-search (BFS). In [1] only a BFS strategy is used to get small graphs.

5 Reducing the Number of QAP Instances

We have strategies to circumvent costly but unavailing QAP computations.

5.1 Padberg-Rinaldi Shrinking Criteria

A vertex set S in a TSP support graph is called *shrinkable* if there is a TSP cut after shrinking S given there was one before. In [12] a sufficient condition C for a mincut S to be shrinkable is given. This shrinking is commonly applied for $|S| \leq 3$ in which case C is automatically satisfied. For $|S| > 3$ it is usually not applied since it is harder to find suitable candidates S and C has to be checked. However, all candidates S can be extracted from the cactus representation of all mincuts and the condition C can be checked efficiently for $|S| \leq 7$ exploiting STSP(k) up to $k = 9$. We shrink shrinkable sets (also 1-squares [12]) to reduce redundancy in TSP support graphs.

P-R shrinking can also help to avoid QAP computations. We only generate small TSP support graphs (Sect. 4.1) of order $k + 1$ if we have processed all small k -vertex graphs. We are satisfied with some cuts per small G . Assume there is a shrinkable set S in G . If there are cuts for G , then also for G/S . We have already processed G/S . We can therefore skip G without separation.

5.2 Graph Isomorphism Detection

We maintain a pool of all small graphs for which we could not find a small cut. A weighted small graph which is isomorphic to a graph in this pool can be ignored. This occurs frequently. It saves numerous unsuccessful QAP computations. Our isomorphism test is fast. We enumerate bijections mapping vertices to equivalent vertices. The equivalence relation is crucial. For us, two vertices are equivalent if they have the same set of incident edge weights.

5.3 Facet Class Selection

For $k \leq 9$ we check all facet classes (except for SECs and $x_e \geq 0$) of STSP(k). This is too slow for $k = 10$. An extreme point χ^T in a facet is a *root* of the facet. Facets in the same symmetry class have the same number of roots. We learned from experiments that a) facets with few roots are rarely violated and b) cuts obtained by lifting such facets get quickly non-binding in the LP. We require at least 100 roots yielding exactly 500 classes (Fig. 1). Simple [10] classes cannot be obtained from smaller TSP polytopes by zero-lifting.

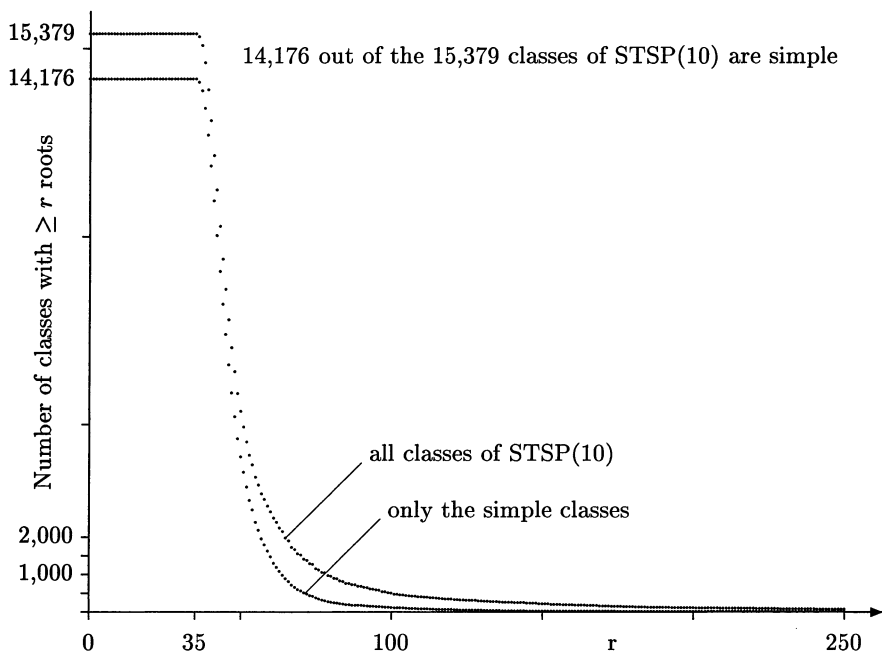


Fig. 1. Data on facet classes of STSP(10) and their number of roots

6 Computational SIR Results

We calculated on a 2800 MHz Pentium 4 processor. The test bed is chosen to allow comparison with [7] where a parallel machine was used. For Table 2 we only produced small graphs as described in Sect. 4.1 and used GRASP-sparse (20 iterations) for the QAP. As [7], we start SIR separation if standard separation (exact SEC, heuristic comb and clique-tree [12]) fails. Standard cuts alone (NT₋) on average result in 70% larger B&C trees than using SIR cuts on top (NT₊). The number of generated different SIR cuts (at most 200 per LP) is listed. In [7] only 5,124 SIR cuts (some duplicates) were generated.

The new SIR approach clearly finds far more cuts. Small facet separation was successful for about 20% of the small graphs actually tested for cuts. P-R criteria (Sect. 5.1) allowed us to skip many (#P-R) small graphs without separation. The isomorphism criterion sorted out #Iso further graphs.

Table 2. Computational results for SIR cuts on top of standard cuts

Instance	NT ₋	NT ₊	Cuts	Tested	Success	#P-R	#Iso
pr152	6	2	1,057	188	71	1,230	50
si175	24	18	3,401	545	220	29,504	275
rat195	28	8	3,327	535	122	4,053	70
d198	10	8	1,423	711	215	10,728	1,302
gr229	30	26	2,691	1,463	194	16,454	405
gil262	14	16	1,149	522	122	10,346	470
pr299	68	32	1,740	1,087	197	27,115	225
lin318	6	2	849	484	93	41,100	118
pr439	74	82	7,557	3,446	663	481,231	2,223
d493	24	24	6,820	1,958	536	89,161	816
att532	102	26	4,995	3,607	540	237,779	1,966
ali535	8	6	1,912	677	187	29,696	367
si535	12	0	778	457	186	10,909	247
gr666	62	32	4,019	3,423	376	41,908	1,991
rat783	16	2	1,019	411	101	6,139	85
si1032	2	2	7	48	1	112	60
Sum	486	286	42,744	19,562	3,824	1,037,465	10,670

7 The SIR Approach Versus Local Cuts

We compare the SIR approach outlined here with the local cut approach (LOC) detailed in [1]. Classical inequality classes are a firm basis for both.

A difference is that SIR follows the paradigm of searching for a cut in a predefined class of inequalities while LOC does not. In theory, LOC computes, by general-purpose techniques, a single cut facet-defining for the graphical TSP (GTSP) polytope if one exists for a given small graph. In practice, LOC may fail even if a cut exists and the computed cut is not necessarily facet-defining for the GTSP polytope. LOC deliberately goes for a single cut and is very fast. SIR, currently restricted to $k \leq 10$, usually computes several facet-defining cuts for STSP(k) if $y^* \notin \text{STSP}(k)$ but may also fail if the involved QAP instances are solved heuristically or facet classes are selected. SIR offers the chance to select ‘good’ cuts from the computed cuts for a small graph.

The partitions chosen by LOC usually contain the vast majority of vertices in a single element. Therefore, a local cut tends to affect an LP solution locally. Our mincut k -partitions often have several elements of large size so that resulting cuts can affect an LP solution more globally.

8 Summary and Conclusion

We have described a way to exploit the facets of small TSP polytopes for cutting-plane generation. The approach outlined here has nothing in common with the pioneering work [6,7] apart from the idea to use the small polytopes to find cuts. The new SIR approach for the TSP finds more cuts. We have suggested strategies to reduce QAP computations – the bottleneck. Our method to enumerate all k -way cuts where each subset is a mincut is interesting beyond the TSP. For the TSP, it seems promising to combine this method to generate small graphs with the local cut ideas by Applegate et al. [1].

References

1. D. Applegate, R. Bixby, V. Chvátal, and W. Cook. TSP Cuts Which Do Not Conform to the Template Paradigm. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*, volume 2241 of *LNCS*, pages 261–303. Springer, 2001.
2. S. C. Boyd and W. H. Cunningham. Small travelling salesman polytopes. *Mathematics of Operations Research*, 16(2):259–271, May 1991.
3. R. E. Burkard and U. Derigs. *Assignment and Matching Problems: Solution Methods with Fortran Programs*, volume 184 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, 1980.
4. T. Christof. SMAPO: A library of linear descriptions of SMALL Polytopes associated with combinatorial optimization problems, 1997. Available at <http://www.informatik.uni-heidelberg.de/groups/comopt>
5. T. Christof, M. Jünger, and G. Reinelt. A complete description of the traveling salesman polytope on 8 nodes. *Operations Research Letters*, 10:497–500, 1991.
6. T. Christof and G. Reinelt. Parallel Cutting Plane Generation for the TSP. In P. Fritzson and L. Finmo, editors, *Parallel Programming and Applications*, pages 163–169. IOS Press, 1995.
7. T. Christof and G. Reinelt. Combinatorial Optimization and Small Polytopes. *Top*, 4(1):1–64, 1996. ISSN 1134-5764.
8. T. Christof and G. Reinelt. Decomposition and Parallelization Techniques for Enumerating the Facets of Combinatorial Polytopes. *International Journal of Computational Geometry & Applications*, 11(4):423–437, 2001.
9. D. R. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, July 1996.
10. D. Naddef and G. Rinaldi. The graphical relaxation: A new framework for the Symmetric Traveling Salesman Polytope. *Math. Program.*, 58(1):53–88, 1993.
11. P. M. Pardalos, L. S. Pitsoulis, and M. G. C. Resende. Algorithm 769: Fortran Subroutines for Approximate Solution of Sparse Quadratic Assignment Problems Using GRASP. *ACM TOMS*, 23(2):196–208, 1997.
12. M. Padberg and G. Rinaldi. Facet identification for the symmetric traveling salesman polytope. *Mathematical Programming*, 47:219–257, 1990.
13. K. M. Wenger. A New Approach to Cactus Construction Applied to TSP Support Graphs. In W. J. Cook and A. S. Schulz, editors, *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, Proceedings*, volume 2337 of *LNCS*, pages 109–126. Springer, 2002.

A Remark on Multiobjective Stochastic Optimization Problems: Stability and Empirical Estimates

Vlasta Kaňková

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Praha 8, Czech Republic
e-mail: kankova@utia.cas.cz

Abstract. We consider a multiobjective optimization problem in which objective functions are in the form of mathematical expectation of functions depending on a random element and a constraints set can depend on the probability measure. Evidently then probability measure can be considered as a parameter of the problem. The aim of this note is to present a survey of some assertions on a stability and statistical estimates of the set of (properly) efficient points. To this end, already known results from one-objective stochastic programming theory are employed.

1 Introduction

It happens rather often that it is reasonable to evaluate an economic activity simultaneously by several “utility” functions. If, moreover, there exist a random element and a “parameter” (not determined completely but whose value must only fulfil some conditions), then a multiobjective optimization problem with a random factor usually corresponds (from the mathematical point of view) to such situation. A rather general multiobjective optimization problem with a random element can be introduced in the following form:

Find

$$\min g_i(x, \xi), \quad i = 1, 2, \dots, l_1 \quad (1)$$

$$\text{subject to} \quad x \in X \quad \text{such that} \quad g_i(x, \xi) \leq 0, \quad i = l_1 + 1, \dots, l. \quad (2)$$

$g_i, i = 1, \dots, l$ are functions defined on $R^n \times R^s$; $\xi := \xi(\omega)$ is an s -dimensional random vector defined on a probability space (Ω, \mathcal{S}, P) ; $X \subset R^n$ is a nonempty set. ($R^n, n \geq 1$ denotes n -dimensional Euclidean space.)

If the solution x can depend on the random element ξ realization, then (1), (2) is a problem of the deterministic (maybe parametric) multiobjective optimization. If x has to be determined without knowing the random element realization, then at first some new decision rule must be selected, i.e. a deterministic optimization problem must be assigned to the original problem (1), (2). This deterministic problem can generally depend on the random element through the “underlying” probability measure and can be one-objective as

well as multiobjective. A different approaches can be employed to assign a new “deterministic” problem. In this note we follow the philosophy of [10] (see also [1], [19]). To this end we assume.

- A.1. It is “reasonable ” to prefer a few of the objective functions, say g_i , $i = 1, \dots, l_2$, $1 \leq l_2 \leq l_1$,
 A.2. there exist constants $k_{l_2+1}, \dots, k_{l_1}$ such that the fulfilling of the relations

$$g_i(x, \xi) \leq k_i, i = l_2 + 1, \dots, l_1, x \in X \text{ is “very” acceptable.} \quad (3)$$

Evidently, under these (very often “acceptable”) assumptions applying the stochastic programming approach we can assign to the problem (1), (2) a “deterministic” problem in the following (general) form:

Find

$$\min E_{F^\xi} \hat{g}_i(x, \xi), \quad i = 1, \dots, l_2 \quad \text{subject to } x \in \mathcal{K}_{F^\xi}, \quad (4)$$

where \hat{g}_i , $i = 1, \dots, l_2$ (defined on $R^n \times R^s$) can correspond to penalty or recourse approach in one-objective stochastic programming problems. The set \mathcal{K}_{F^ξ} can be equal to X or it can correspond to probability constraints. Evidently in the second case \mathcal{K}_{F^ξ} is determined by the systems (2), (3) and depends on the distribution function F^ξ of ξ . However, everywhere $\mathcal{K}_{F^\xi} \subset X$ (for more details see e.g. [10] or [14]).

The problem introduced by (4) depends on the distribution function F^ξ . Consequently, F^ξ (or equivalently the corresponding probability measure P_{F^ξ}) can be considered as a parameter of the problem. The aim of this note is, first, to deal with the stability with respect to the distribution function parameter. (Of course, to investigate the stability of the problem (4) it means to investigate the stability of the efficient points set.) Furthermore, we intend to deal with the case when an empirical measure substitutes the P_{F^ξ} .

To obtain new results for multiobjective stochastic problems we employ the results achieved for one-objective problems. Namely, a great attention has been paid to the stability and empirical estimates in one-objective stochastic programming literature (see e.g. [3], [7], [8], [17], [20]).

2 Some Definitions and Auxiliary Assertions

A multiobjective deterministic optimization problem can be introduced as the problem:

Find

$$\min f_i(x), \quad i = 1, \dots, r \quad \text{subject to } x \in \mathcal{K}. \quad (5)$$

f_i , $i = 1, \dots, r$ are functions defined on R^n , $\mathcal{K} \subset R^n$ is a nonempty set.

Definition 1. The vector x^* is an efficient solution of the problem (5) if and only if there exists no $x \in \mathcal{K}$ such that $f_i(x) \leq f_i(x^*)$ for $i = 1, \dots, r$ and such that for at least one i_0 one has $f_{i_0}(x) < f_{i_0}(x^*)$.

Definition 2. The vector x^* is a properly efficient solution of the multiobjective optimization problem (5) if and only if it is efficient and if there exists a scalar $\bar{M} > 0$ such that for each i and each $x \in \mathcal{K}$ satisfying $f_i(x) < f_i(x^*)$ there exists at least one j such that $f_j(x^*) < f_j(x)$ and

$$\frac{f_i(x^*) - f_i(x)}{f_j(x) - f_j(x^*)} \leq \bar{M}. \quad (6)$$

Proposition 1. [5] Let \mathcal{K} be a convex set and let $f_i, i = 1, \dots, r$ be convex functions on \mathcal{K} . Then x^0 is a properly efficient solution of the problem (5) if and only if x^0 is optimal in

$$\min_{x \in \mathcal{K}} \sum_{i=1}^r \lambda_i f_i(x) \quad \text{for some } \lambda_1, \dots, \lambda_r > 0; \quad \sum_{i=1}^r \lambda_i = 1.$$

Definition 3. Let h be a function defined on a convex set $\mathcal{K} \subset \mathbb{R}^n$. h is a strongly convex function with a parameter $\rho > 0$ if

$$h(\lambda x^1 + (1 - \lambda)x^2) \leq \lambda h(x^1) + (1 - \lambda)h(x^2) - \lambda(1 - \lambda)\rho \|x^1 - x^2\|^2$$

for every $x^1, x^2 \in \mathcal{K}$, $\lambda \in (0, 1)$.

Lemma 1. [7] Let \mathcal{K} be a nonempty, compact, convex set. Let, moreover, h be a strongly convex with a parameter $\rho > 0$ continuous function defined on \mathcal{K} . If x^0 is defined by the relation $x^0 = \arg \min_{x \in \mathcal{K}} h(x)$, then

$$\|x - x^0\|^2 \leq \frac{2}{\rho} |h(x) - h(x^0)| \quad \text{for every } x \in \mathcal{K}.$$

($\|\cdot\|$ denotes the Euclidean norm in \mathbb{R}^n .)

If F and G are two arbitrary s -dimensional distribution functions, then the Kolmogorov metric $d_K(P_F, P_G)$ is defined by

$$d_K(P_F, P_G) := d_K(F, G) = \sup_{z \in \mathbb{R}^s} |F(z) - G(z)|. \quad (7)$$

To define the Wasserstein metric $d_{W_1}(P_F, P_G) := d_{W_1}(F, G)$ let $\mathcal{P}(\mathbb{R}^s)$ denote the set of (all) Borel probability measures on \mathbb{R}^s , $P_F, P_G \in \mathcal{P}(\mathbb{R}^s)$ and let

$$\mathcal{M}_1(\mathbb{R}^s) = \{\nu \in \mathcal{P}(\mathbb{R}^s) : \int_{\mathbb{R}^s} \|z\| \nu(dz) < \infty\}.$$

If $\mathcal{D}(\nu, \mu)$ denotes the set of those measures in $\mathcal{P}(\mathbb{R}^s \times \mathbb{R}^s)$ whose marginal measures are ν and μ , then

$$d_{W_1}(\nu, \mu) = \inf \left\{ \int_{\mathbb{R}^s \times \mathbb{R}^s} \|z - \bar{z}\| \kappa(dz \times d\bar{z}) : \kappa \in \mathcal{D}(\nu, \mu) \right\}, \quad \nu, \mu \in \mathcal{M}_1(\mathbb{R}^s). \quad (8)$$

(For more details about the Wasserstein metric see e.g. [15].)

3 Main Results

To study the stability of the problem (4) we define the sets \mathcal{G}_{F^ξ} , \mathcal{X}_{F^ξ} and $\bar{\mathcal{G}}_{F^\xi}$. To this end we denote the elements of R^{l_2} by $y = (y_1, \dots, y_{l_2})$.

$$\begin{aligned}\mathcal{G}_{F^\xi} &= \{y \in R^{l_2} : y_j = E_{F^\xi} \hat{g}_j(x, \xi), j = 1, \dots, l_2 \text{ for some } x \in \mathcal{K}_{F^\xi}\}, \\ \mathcal{X}_{F^\xi} &= \{x \in X : x \text{ is a properly efficient point of the problem (4)}\},\end{aligned}\tag{9}$$

$$\bar{\mathcal{G}}_{F^\xi} = \{y \in R^{l_2} : y_j = E_{F^\xi} \hat{g}_j(x, \xi), j = 1, \dots, l_2 \text{ for some } x \in \mathcal{X}_{F^\xi}\}.$$

If we replace in (4) F^ξ by another s -dimensional distribution function G , then we obtain \mathcal{K}_G instead of \mathcal{K}_{F^ξ} in (4) and \mathcal{G}_G , \mathcal{X}_G and $\bar{\mathcal{G}}_G$ instead of \mathcal{G}_{F^ξ} , \mathcal{X}_{F^ξ} , $\bar{\mathcal{G}}_{F^\xi}$ in (9). The aim of this section is, first, to investigate

$$\Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_G], \quad \Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_G], \quad \Delta[\bar{\mathcal{G}}_{F^\xi}, \bar{\mathcal{G}}_G],$$

where $\Delta[\cdot, \cdot]$ denotes the Hausdorff distance in the space of nonempty, closed subsets of R^n (for the definition of the Hausdorff distance see e.g. [18]). To this end we employ the Kolmogorov and the Wasserstein metrics. The bounded Lipschitz metric β has been employed to investigate stability of multiobjective stochastic linear problems [2]; the results of [16] was employed there. If we replace F^ξ by its empirical estimate F_N^ξ , then we obtain empirical estimates of \mathcal{G}_{F^ξ} , \mathcal{X}_{F^ξ} and $\bar{\mathcal{G}}_{F^\xi}$. We intend to investigate these estimates by a large deviation technique [6].

3.1 Stability Results

To introduce the stability results let us, first, denote the support of the probability measure P_{F^ξ} by Z_{F^ξ} and let for $\delta > 0$, $Z_{F^\xi}(\delta)$ denote the δ -neighborhood of the set Z_{F^ξ} . We introduce the following assumptions.

- B.1 a. $\hat{g}_i, i = 1, \dots, l_2$ are uniformly continuous functions on $X \times R^s$,
 b. for every $x \in X$, $\hat{g}_i, i = 1, \dots, l_2$ are Lipschitz functions of $z \in R^s$ with the Lipschitz constants not depending on $x \in X$.
- B.2 a. $\hat{g}_i, i = 1, \dots, l_2$ are Lipschitz functions on $X(3\delta) \times Z_F(\sqrt{s}\delta)$, (with the Lipschitz constants L_i),
 b. X is a convex set and simultaneously $\hat{g}_i, i = 1, \dots, l_2$ are strongly convex (with a parameter $\rho > 0$) functions on $X(3\delta)$,
 c. \mathcal{K}_{F^ξ} is a nonempty, convex, compact set.
- B.3 a. P_{F^ξ} is absolutely continuous w.r.t. the Lebesgue measure on R^s . (We denote by h^ξ the probability density corresponding to P_{F^ξ}),
 b. $Z_{F^\xi} = \bigcap_{j=1}^s \langle c_j, c'_j \rangle$, $c_j, c'_j > 0$, $c_j < c'_j$, $j = 1, 2, \dots, s$ and, moreover, there exists a constant $\vartheta > 0$ such that $h^\xi(z) \geq \vartheta$ for every $z \in Z_{F^\xi}$.

Theorem 1. Let $\delta > 0$, \mathcal{K}_{F^ξ} be a nonempty compact set, $P_{F^\xi} \in \mathcal{P}(R^s)$. If

1. the assumptions B.2a, B.3 are fulfilled,
2. $G \in \mathcal{P}(R^s)$ is an arbitrary s -dimensional distribution function such that

$$Z_G \subset Z_{F^\xi}(\delta') \quad \text{for} \quad \delta' = \left(\frac{2d_K(P_{F^\xi}, P_G)}{\vartheta} \right)^{\frac{1}{s}} \leq \min_j [\min(c'_j - c_j), \delta].$$

then there exists a constant $C'_{\mathcal{G}_{F^\xi}} > 0$ such that

$$\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq k[d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}} \leq \delta, \quad X_G \subset X \text{ a nonempty, compact set}$$

$$k > 0 \implies \Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_G] \leq C'_{\mathcal{G}_{F^\xi}} [d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}}.$$

If, moreover, the assumptions B.2b, B.2c are fulfilled, then there exist constants $C'_{\mathcal{X}_{F^\xi}}, C'_{\bar{\mathcal{G}}_{F^\xi}}, K'_{\bar{\mathcal{G}}_{F^\xi}} > 0$ such that the following implications hold

$$\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq kd_K(P_{F^\xi}, P_G) \leq \delta, \quad k > 0, \quad X_G \subset X \text{ a nonempty,}$$

$$\text{convex, compact set} \implies \Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_G]^2 \leq C'_{\mathcal{X}_{F^\xi}} [d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}},$$

$$\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq kd_K(P_{F^\xi}, P_G) \leq \delta, \quad X_G \subset X \text{ a nonempty, convex, compact}$$

$$k > 0 \implies \Delta[\bar{\mathcal{G}}_{F^\xi}, \bar{\mathcal{G}}_{F^\xi}] \leq C'_{\bar{\mathcal{G}}_{F^\xi}} \sqrt{[d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}} (K'_{\bar{\mathcal{G}}_{F^\xi}} + \sqrt{[d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}}}}.$$

The sketch of the proof of Theorem 1 will be given in the Appendix.

To the validity of Theorem 1 the inequality $\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq kd_K(P_{F^\xi}, P_G)$ for some $k > 0$ must be verified. To this end we can recall the results of [9]. There the case when \mathcal{K}_{F^ξ} corresponds to the probability constraints and $l - l_2 = s$, $g_{i+l_2}(x, z) = f_i(x) - z_i$ for some functions f_i defined on R^n , $i = 1, \dots, s$, $z = (z_1, \dots, z_s)$ is investigated. If $\mathcal{K}_{F^\xi} = X' \subset X$ independently on F^ξ , then the inequality $\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_G] \leq kd_K(P_{F^\xi}, P_G)$ is fulfilled every time. However, in this special case the Wasserstein metric can be also employed.

Theorem 2. Let $P_{F^\xi} \in \mathcal{M}_1(R^s)$, the system of the assumptions B.1 be fulfilled. Let, moreover, $X \subset R^n$ be a nonempty, compact set. If

1. $\mathcal{K}_{F^\xi} = X$ independently on F^ξ ,
2. G is an arbitrary s -dimensional distribution function; $P_G \in \mathcal{M}_1(R^s)$,

then there exists a constant $C''_{\mathcal{G}_{F^\xi}} > 0$ such that

$$\Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_G] \leq C''_{\mathcal{G}_{F^\xi}} d_{W_1}(P_{F^\xi}, P_G).$$

If, moreover,

- a. the assumptions B.2b, B.2c are fulfilled, then there exists a constant $C''_{\mathcal{X}_{F^\xi}} > 0$ such that

$$\Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_G]^2 \leq C''_{\mathcal{X}_{F^\xi}} d_{W_1}(P_{F^\xi}, P_G),$$

- b. the system of the assumptions B.2 is fulfilled, then there exist constants $C''_{\bar{G}_{F^\xi}}, K''_{\bar{G}_{F^\xi}} > 0$ such that

$$\Delta[\bar{G}_{F^\xi}, \bar{G}_G] \leq C''_{\bar{G}_{F^\xi}} \sqrt{d_{W_1}(P_{F^\xi}, P_G)} (K''_{\bar{G}_{F^\xi}} + \sqrt{d_{W_1}(P_{F^\xi}, P_G)}).$$

The sketch of the proof of Theorem 2 will be given in the Appendix.

3.2 Empirical Estimates

To deal with empirical estimates, let $\{\xi^k\}_{k=1}^N$, $N = 1, 2, \dots$ be a sequence of independent s -dimensional random vectors with a common distribution function F^ξ . We denote by the symbol F_N^ξ the empirical distribution function determined by $\{\xi^k\}_{k=1}^N$.

Theorem 3. Let X, \mathcal{K}_{F^ξ} be a nonempty, compact sets, $t > 0$. If

1. $\hat{g}_i, i = 1, \dots, l_2$
 - a. are uniformly continuous, bounded functions on $X \times Z_{F^\xi}$,
 - b. for every $z \in Z_{F^\xi}$, Lipschitz functions on X with the Lipschitz constants not depending on $z \in Z_{F^\xi}$,
2. there exist constants $K^1 := K^1(X, t) > 0, k^1 > 0$ such that

$$P\{\Delta[\mathcal{K}_{F^\xi}, \mathcal{K}_{F_N^\xi}] > t\} \leq K^1(X, t) \exp\{-k^1 N t^2\}, \quad N = 1, 2, \dots$$

then there exist constants $K_{\mathcal{G}_{F^\xi}}^1 := K_{\mathcal{G}_{F^\xi}}^1(X, t) > 0, k_{\mathcal{G}_{F^\xi}}^1 > 0$ such that

$$\mathcal{K}_{F_N^\xi} \subset X, P\{\Delta[\mathcal{G}_{F^\xi}, \mathcal{G}_{F_N^\xi}] > t\} \leq K_{\mathcal{G}_{F^\xi}}^1(X, t) \exp\{-k_{\mathcal{G}_{F^\xi}}^1 N t^2\}.$$

If, moreover,

3. X, \mathcal{K}_{F^ξ} are convex sets,
4. $\hat{g}_i, i = 1, 2, \dots, l_2$ are strongly convex (with a parameter $\rho > 0$) functions on X ,

then there exist constants $K_{\mathcal{X}_{F^\xi}}^1 := K_{\mathcal{X}_{F^\xi}}^1(X, t) > 0, k_{\mathcal{X}_{F^\xi}}^1 > 0$ such that

$$P\{[\Delta[\mathcal{X}_{F^\xi}, \mathcal{X}_{F_N^\xi}]]^2 > t\} \leq K_{\mathcal{X}_{F^\xi}}^1(X, t) \exp\{-k_{\mathcal{X}_{F^\xi}}^1 N t^2\}, \quad N = 1, 2, \dots$$

The sketch of the proof of Theorem 3 will be given in the Appendix.

Appendix

Sketch of the proof of Theorem 1. First, employing the proof technique of [9] we can get that there exists a constant $K > 0$ such that

$$|\mathbb{E}_{F^\xi} \hat{g}_i(x, \xi) - \mathbb{E}_G \hat{g}_i(x, \xi)| \leq [K d_K(P_{F^\xi}, P_G)]^{\frac{1}{s}}, \quad i = 1, \dots, l_2, \quad x \in X.$$

Furthermore employing the triangular inequality and the properties of the Hausdorff distance we obtain the first assertion of Theorem 1.

To obtain the others assertions we define for $\lambda_i \in (0, 1)$, $i = 1, \dots, l_2$, $\sum_{i=1}^{l_2} \lambda_i = 1$, the function $g_{\bar{\lambda}}$, $\bar{\lambda} = (\lambda_1, \dots, \lambda_{l_2})$ by

$$\hat{g}_{\bar{\lambda}}(x, z) = \sum_{i=1}^{l_2} \lambda_i \hat{g}_i(x, z), \quad x \in R^n, z \in R^s.$$

It is easy to see that (under the assumptions) $\hat{g}_{\bar{\lambda}}$ is a Lipschitz, strongly convex (with a parameter not depending on $\bar{\lambda}$) function on $X(3\delta) \times Z_F(\sqrt{s}\delta)$ with the Lipschitz constant not greater then $\sum_{i=1}^l L_i$ (independently on $\bar{\lambda} = (\lambda_1, \dots, \lambda_{l_2})$).

Employing furthermore the triangular inequality we can obtain

$$\begin{aligned} & \left| \inf_{\mathcal{K}_{F\xi}} E_{F\xi} \hat{g}_{\bar{\lambda}}(x, \xi) - \inf_{\mathcal{K}_G} E_G \hat{g}_{\bar{\lambda}}(x, \xi) \right| \leq \left| \inf_{\mathcal{K}_{F\xi}} E_{F\xi} \hat{g}_{\bar{\lambda}}(x, \xi) - \inf_{\mathcal{K}_{F\xi}} E_G \hat{g}_{\bar{\lambda}}(x, \xi) \right| + \\ & \left| \inf_{\mathcal{K}_{F\xi}} E_G \hat{g}_{\bar{\lambda}}(x, \xi) - \inf_{\mathcal{K}_G} E_G \hat{g}_{\bar{\lambda}}(x, \xi) \right|. \end{aligned} \quad (10)$$

The proven assertions follows now already from the last inequality, the assertion of Proposition 1, the stability assertions for one-objective problems [9], Lemma 1 and the properties of the Hausdorff distance.

Sketch of the proof of Theorem 2. The proof of Theorem 2 is similar to the proof of Theorem 1. However instead of the inequalities for the Kolmogorov metric [9] the results for Wasserstein metric [12] must be employed.

The complete proofs of Theorem 1 and Theorem 2 are given in [13].

Sketch of the proof of Theorem 3. To prove the first assertion of Theorem 3 we employ the approach of [7] or [8] to get for every $t > 0$ the existence of constants $\bar{k}_1^* > 0$, $K_1^*(X, t) > 0$ such that

$$\begin{aligned} & P\{|\mathbb{E}_{F_N^\xi} \hat{g}_i(x, \xi) - \mathbb{E}_{F_N^\xi} \hat{g}_i(x, \xi)| > t \text{ for at least one } x \in X \\ & \text{and at least one } i \in \{1, \dots, l_2\}\} \leq K_1^*(X, t) \exp\{-\bar{k}_1^* N t^2\}, \quad N = 1, 2, \dots \end{aligned}$$

To prove the second assertion, we employ again the assertion of Proposition 1 and Lemma 1, the triangular inequality, the properties of the Hausdorff distance and the results of [7] or [8].

The complete proof of Theorem 3 (for the case when $\mathcal{K}_{F\xi}$ correspond to the individual probability constraints) is given in [11].

Acknowledgement. This research was supported by the Grant Agency of the Czech Republic under Grants 402/02/1015 and 402/01/0539.

References

1. Caballero, R., Cerdá, E., Muñoz, M. M., Rey, L., Stancu-Minasian, I. M. (2001): Efficient solution concepts and their relations in stochastic multiobjective programming. *J. Optim. Theory Appl.*, 110, 1, 53–74
2. Cho, G.-M. (1995): Stability of the multiple objective linear stochastic programming problems. *Bull. Korean Math. Soc.*, 32, 2, 287–296
3. Dupačová, J., Wets, R. J.-B. (1984): Asymptotic behaviour of statistical estimates and optimal solutions of stochastic optimization problems. *Ann. Statist.*, 16, 1517–1549
4. Dupačová, J., Hurt, J., Štěpán, J. (2002): *Stochastic Modeling in Economics and Finance*. Kluwer, Dordrecht
5. Geoffrion, A. M. (1968): Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.*, 22, 3, 618–630
6. Hoeffding, W. (1963): Probability inequalities for sums of bounded random variables. *J. Amer. Statist.*, 58, 301, 13–30
7. Kaňková, V., Lachout, P. (1992): Convergence rate of empirical estimates in stochastic programming. *Informatica*, 3, 4, 497–523
8. Kaňková, V. (1994): A note on estimates in stochastic programming. *J. Comp. Math.*, 56, 97–112
9. Kaňková, V. (1998a): A note on multifunctions in stochastic programming. In: *Stochastic Programming Methods and Technical Applications* (K. Marti, P. Kall, eds.). Springer, Berlin
10. Kaňková, V. (1998b): A note on analysis of economic activities with random elements. In: *Mathematical Methods in Economy, Cheb 1998* (M. Plevný and V. Friedrich, eds.) University of West Bohemia, Cheb, 53–58
11. Kaňková, V. (2000): *Stochastic Programming Approach to Multiobjective Optimisation Problems with Random Elements I*. Research Report UTIA AS CR No. 1990
12. Kaňková, V., Houda, M. (2002): A note on quantitative stability and empirical estimates in stochastic programming. In: *Operations Research Proceedings 2002*, Springer, Berlin, 413–418
13. Kaňková, V. (2002): *Stability in Multiobjective Stochastic Programming Problems*. Research Report UTIA AS CR No. 1990
14. Prékopa, A. (1995): *Stochastic Programming*. Akadémiai Kiadó, Budapest and Kluwer, Dordrecht
15. Rachev, S. T. (1991): *Probability Metrics and the Stability of Stochastic Models*. Wiley, Chichester
16. Römisch, W., Wakolbinger, A. (1987): Obtaining convergence rates for approximations in stochastic programming. In: *Parametric Optimization and Related Topics* (J. Guddat, H. Th. Jongen, B. Kummer, F. Nožička, eds.), Akademie, Berlin, 327–343
17. Römisch, W., Schulz, R. (1993): Stability of solutions for stochastic programs with complete recourse. *Math. Oper. Res.*, 18, 590–609
18. Salinetti, G., Wets, R. J.-B. (1979): On the convergence of sequences of convex sets in finite dimension. *SIAM Review*, 21, 18–33
19. Stancu-Minasian, I. M. (1984): *Stochastic Programming with Multiple Objective Functions*. D. Reidel Publishing Company, Dordrecht
20. Vogel, S. (1992): On stability in multiobjective programming – a stochastic approach. *Mathematical Programming*, 50, 197–236

Portfolio Optimization under Partial Information: Stochastic Volatility in a Hidden Markov Model^{*}

Jörn Sass¹ and Ulrich G. Haussmann²

¹ RICAM, Altenberger Str. 69, A-4040 Linz, Austria, joern.sass@oeaw.ac.at

² Math. Dept., UBC, Vancouver, BC, V6T 1Z2, Canada, uhaus@math.ubc.ca

Abstract. We consider a multi-stock market model where prices satisfy a stochastic differential equation (SDE) with instantaneous rates of return modeled as an unobserved continuous time, finite state Markov chain. The investor wishes to maximize the expected utility of terminal wealth but only the prices are available to him for his investment decisions. Thus we have a hidden Markov model (HMM) for the stock returns. Extending the results in [9] to stochastic volatility we obtain explicit optimal trading strategies in terms of the unnormalized filter of the drift process. We propose a simple volatility model in which the volatility is a function of the filter for the drift process. When applied to historical prices, the optimal strategies clearly outperform the strategies based on constant volatility.

1 Outline

In Sect. 2 we extend the HMM of [9] by introducing a complete model with stochastic volatility. While the HMM with constant volatility already improves the Merton strategy, considering the stochastic volatility model proposed in Example 1 leads to even better optimization results, for simulated as well as for market data (Sect. 5). Our main contributions are the extension of Theorem 2 to stochastic volatility and the implementation of a new filter-dependent practical volatility model. An algorithm for the parameter estimation is presented in Sect. 4. To keep our notation simple we assume interest rates equal 0. For stochastic interest rates, motivation of the model, Malliavin calculus, further aspects of parameter estimation, and for more references see [9].

Notational remarks: T will denote transposition, $\text{Diag}(v)$ the diagonal matrix with diagonal v , $\mathcal{F}^X = (\mathcal{F}_t^X)_{t \in [0, T]}$ the augmented σ -algebra generated by an \mathcal{F} -adapted process X .

^{*} Supported by NSERC under research grant 88051 and NCE grant 30354 and by the Austrian Academy of Sciences.

2 An HMM for the Stock Returns

Let (Ω, \mathcal{A}, P) be a complete probability space, $T > 0$ the terminal trading time, and $\mathcal{F} = (\mathcal{F}_t)_{t \in [0, T]}$ a filtration in \mathcal{A} satisfying the usual conditions. We consider one *money market* with interest rates equal 0 and n stocks whose *prices* $S = (S_t)_{t \in [0, T]}$, $S_t = (S_t^1, \dots, S_t^n)^\top$ evolve according to

$$dS_t = \text{Diag}(S_t)(\mu_t dt + \sigma_t dW_t), \quad S_0 \in \mathbb{R}^n,$$

where $W = (W_t)_{t \in [0, T]}$ is an n -dimensional Brownian motion w.r.t. \mathcal{F} and the $(n \times n)$ -volatility-matrices $(\sigma_t)_{t \in [0, T]}$ are uniformly bounded, progressively measurable w.r.t. \mathcal{F} and non-singular with uniformly bounded $(\sigma_t^{-1})_{t \in [0, T]}$. The *return process* $R = (R_t)_{t \in [0, T]}$ is defined by $dR_t = (\text{Diag}(S_t))^{-1} dS_t = \mu_t dt + \sigma_t dW_t$. We assume that $\mu = (\mu_t)_{t \in [0, T]}$, the *drift process* of the return, is given by $\mu_t = B Y_t$, $t \in [0, T]$, where $Y = (Y_t)_{t \in [0, T]}$ is a stationary, irreducible, *continuous time Markov chain* independent of W with state space $\{e_1, \dots, e_d\}$, the standard unit vectors in \mathbb{R}^d , and the columns of the *state matrix* $B \in \mathbb{R}^{n \times d}$ contain the d possible states of μ_t . Further Y is characterized by its *rate matrix* $Q \in \mathbb{R}^{d \times d}$, where $\lambda_k = -Q_{kk} = \sum_{l=1, l \neq k}^d Q_{kl}$ is the rate of leaving e_k and Q_{kl}/λ_k is the probability that the chain jumps to e_l when leaving e_k .

Since the *market price of risk*, $\vartheta_t = \sigma_t^{-1} B Y_t$, $t \in [0, T]$, is uniformly bounded the density process $(Z_t)_{t \in [0, T]}$, $dZ_t = -Z_t \vartheta_t^\top dW_t$, $Z_0 = 1$, is a martingale. By $d\tilde{P} = Z_T dP$ we define the *risk-neutral* probability measure \tilde{P} which we need for the optimization and which is also the *reference measure* used in filtering. $\tilde{\mathbb{E}}$ will denote expectation with respect to \tilde{P} . Girsanov's Theorem guarantees that $d\tilde{W}_t = dW_t + \vartheta_t dt$ defines a \tilde{P} -Brownian motion. By the definition of R ,

$$R_t = \int_0^t B Y_s ds + \int_0^t \sigma_s dW_s = \int_0^t \sigma_s d\tilde{W}_s, \quad t \in [0, T]. \quad (1)$$

Now we can specify a Markovian model for the dynamics of σ : Let the m -dimensional process $\xi = (\xi_t)_{t \in [0, T]}$ be given by

$$d\xi_t = \nu(\xi_t) dt + \tau(\xi_t) d\tilde{W}_t, \quad (2)$$

where ν , τ are \mathbb{R}^m - and $\mathbb{R}^{m \times n}$ -valued and continuously differentiable with bounded partial derivatives. Furthermore let $\sigma_t = \sigma(\xi_t)$, $t \in [0, T]$, where σ is continuously differentiable with bounded partial derivatives.

The conditions imply the usual Lipschitz and growth conditions, hence guarantee the existence of a strong solution.

3 Optimal Trading Strategies

We consider the case of *partial information* meaning that an investor can only observe the prices. Neither the drift process nor the Brownian motion

are observable. Therefore only the events of \mathcal{F}^S can be observed and all investment decisions have to be adapted to \mathcal{F}^S . Note that $\mathcal{F}^S = \mathcal{F}^R = \mathcal{F}^{\tilde{W}}$.

A trading strategy $\pi = (\pi_t)_{t \in [0, T]}$ is an n -dimensional \mathcal{F}^S -adapted, measurable process which satisfies $\int_0^T \|\sigma_t^\top \pi_t\|^2 dt < \infty$, $\int_0^T |\mu_t^\top \pi_t| dt < \infty$. For initial capital $x_0 > 0$ the corresponding wealth process $X^\pi = (X_t^\pi)_{t \in [0, T]}$ is defined by $dX_t^\pi = \pi_t^\top (\mu_t dt + \sigma_t dW_t)$, $X_0 = x_0$. π_t is the wealth invested in the stock at time t , $X_t - \mathbf{1}_n^\top \pi_t$ is invested in the money market. A trading strategy π is called *admissible* if $P(X_t^\pi \geq K^\pi \text{ for all } t \in [0, T]) = 1$ for some constant $K^\pi > -\infty$. By Itô's rule

$$X_t^\pi = x_0 + \int_0^t \pi_s^\top \sigma_s d\tilde{W}_s, \quad t \in [0, T]. \quad (3)$$

A utility function $U : [x_u, \infty) \rightarrow \mathbb{R} \cup \{-\infty\}$, $x_u \in \mathbb{R}$ is strictly increasing, strictly concave, twice continuously differentiable on (x_u, ∞) , and its derivative U' satisfies $\lim_{x \rightarrow \infty} U'(x) = 0$ and $\lim_{x \rightarrow x_u+} U'(x) = \infty$. We denote the inverse function of U' by $I : (0, \infty) \rightarrow (x_u, \infty)$. We may extend U to $x < x_u$ as $U(x) = -\infty$. The optimization problem is:

$$V_U = \sup \{ E[U(X_T^\pi)] \mid \pi \text{ admissible} \},$$

i.e. find an admissible trading strategy $\hat{\pi}$ which satisfies $E[U(X_T^{\hat{\pi}})] = V_U$. We call $\hat{\pi}$ the *optimal trading strategy*, and denote $\hat{X} = X^{\hat{\pi}}$, so \hat{X}_T is the *optimal terminal wealth*.

To determine the optimal trading strategy we have to find a good estimator for the drift process. By (1) we are in the classical situation of *HMM filtering* with signal Y and observation R , see [1], where we want to determine the filter $E[Y_t \mid \mathcal{F}_t^R]$ for Y_t ; it is an L^2 -optimal estimator.

Theorem 1. The filter $\eta_t = E[Y_t \mid \mathcal{F}_t^S]$ (for Y) the unnormalized filter $\mathcal{E}_t = \tilde{E}[Z_T^{-1} Y_t \mid \mathcal{F}_t^S]$ (for Y), and the conditional density $\zeta_t = E[Z_t \mid \mathcal{F}_t^S]$ (filter for Z), $\zeta_t^{-1} = 1/\zeta_t$, satisfy $\eta_t = \zeta_t \mathcal{E}_t$, $\zeta_t^{-1} = \mathbf{1}_d^\top \mathcal{E}_t$, and

$$\mathcal{E}_t = E[Y_0] + \int_0^t Q^\top \mathcal{E}_s ds + \int_0^t \text{Diag}(\mathcal{E}_s) B^\top (\sigma_s \sigma_s^\top)^{-1} dR_s, \quad t \in [0, T].$$

Proof. The first equation is Bayes' Law. It and $\mathbf{1}_d^\top Y_t = 1$ imply the second. The SDE for \mathcal{E} follows from an extension of Theorem 4 in [1] to stochastic volatility (a direct proof can be given).

The optimal strategy in our main result, Theorem 2, will be expressed in terms of the unnormalized filter \mathcal{E} and its Malliavin derivatives. All quantities in the representation below are observable and the filters and derivatives can be approximated very well because of the linear structure of the equations. For the special use of Malliavin calculus (w.r.t. \tilde{W}) we refer to [9, Sect. 8] and the references therein.

Theorem 2. If $\tilde{\mathbb{E}}[I(x\zeta_T)] < \infty$ for all $x \in (0, \infty)$ and if $I'(\hat{y}\zeta_T) \in L^q(\tilde{P})$ for some $q > 1$, then $\hat{X}_T = I(\hat{y}\zeta_T)$ and for $t \in [0, T]$, $\hat{\pi}_t$ equals

$$\hat{y}^{-1}(\sigma_t \sigma_t^\top)^{-1} \left\{ B \mathcal{E}_t \tilde{\mathbb{E}}[\psi(\hat{y}\zeta_T) | \mathcal{F}_t^S] + \sigma_t \tilde{\mathbb{E}} \left[\psi(\hat{y}\zeta_T) \int_t^T D_t(A(\xi_s) \mathcal{E}_s) d\tilde{W}_s \mid \mathcal{F}_t^S \right] \right\},$$

where $\psi(y) = -y^2 I'(y)$, \hat{y} is uniquely determined by $\tilde{\mathbb{E}}[I(\hat{y}\zeta_T)] = x_0$, $A(x) = (\sigma(x))^{-1} B$, $D_t(A(\xi_s) \mathcal{E}_s) = (D_t \mathcal{E}_s) A(\xi_s)^\top + (D_t \xi_s)(\partial_x A(\xi_s) \mathcal{E}_s)^\top$, and ∂_x denotes the Jacobi matrix w.r.t. the components of ξ . Writing $a_j = (A_j)^\top$

$$\begin{aligned} D_t \mathcal{E}_u &= \sigma_t^{-1} B \text{Diag}(\mathcal{E}_t) + \int_t^u (D_t \mathcal{E}_s) Q ds \\ &\quad + \sum_{j=1}^n \int_t^u \left((D_t \mathcal{E}_s) \text{Diag}(a_j(\xi_s)) + (D_t \xi_s) \text{Diag}(\mathcal{E}_s) (\partial_x a_j(\xi_s))^\top \right) d\tilde{W}_s^j, \\ D_t \xi_u &= (\tau(\xi_t))^\top + \int_t^u (D_t \xi_s) (\partial_x \nu(\xi_s))^\top ds + \sum_{j=1}^n \int_t^u (D_t \xi_s (\partial_x \tau_{\cdot j}(\xi_s))^\top d\tilde{W}_s. \end{aligned}$$

Proof. A straightforward extension of Theorem 2.5 in [7] yields the existence of $\hat{\pi}$ and $\hat{X}_T = I(\hat{y}\zeta_T)$. By comparing (3) and Clark's formula, see [6],

$$\sigma_t^\top \hat{\pi}_t = \tilde{\mathbb{E}}[D_t(\hat{X}_T) \mid \mathcal{F}_t^S]. \quad (4)$$

While the representation for $D_t \xi_s$ follows directly from [9, Prop. 8.2], the equation for \mathcal{E} is more difficult to derive than for constant σ since the required boundedness conditions don't have to hold. But considering $\mathcal{E}_t^N = h(\mathcal{E}_t^N)$ for a bounded function h with suitably bounded partial derivatives, applying [9, Prop. 8.2] to these and using convergence results lead to the stated representation, cf. [3]. The chain rules [9, Props. 8.3, 8.4] yield the equation for $D_t(A(\xi_s) \mathcal{E}_s)$ and that \hat{X}_T lies in the domain $D_{1,1}$ with Malliavin derivative

$$D_t \hat{X}_T = \hat{y} I'(\hat{y}\zeta_T) D_t \zeta_T = -\hat{y} I'(\hat{y}\zeta_T) (\zeta_T)^2 D_t \zeta_T^{-1}. \quad (5)$$

The result follows using (4), (5), the definition of ψ , $Q \mathbf{1}_d = 0$, and

$$D_t \zeta_T^{-1} = (D_t \mathcal{E}_T) \mathbf{1}_d = \sigma_t^{-1} B \mathcal{E}_t + \int_t^T D_t(\sigma_t^{-1} B \mathcal{E}_s) d\tilde{W}_s.$$

Important utility functions are the *logarithmic utility* $U_0(x) = \log(x)$, $x > 0$, $U(0) = -\infty$, and the *power utility* $U_\alpha(x) = x^\alpha/\alpha$, $x \geq 0$, for $\alpha < 1$, $\alpha \neq 0$, because these cover the whole range of risk behaviour. Here $1 - \alpha$ is the Arrow-Pratt index of risk aversion. Analogous to [9, Prop. 4.10],

Corollary 1. Let $\alpha < 1$, $U = U_\alpha$. Then

$$\begin{aligned} \hat{\pi}_t &= \frac{\hat{X}_t(\sigma_t \sigma_t^\top)^{-1}}{(1 - \alpha) \tilde{\mathbb{E}}[\zeta_{t,T}^{\frac{1}{1-\alpha}} \mid \xi_t, \mathcal{E}_t]} \left\{ B \eta_t \tilde{\mathbb{E}} \left[\zeta_{t,T}^{\frac{\alpha}{1-\alpha}} \mid \xi_t, \mathcal{E}_t \right] \right. \\ &\quad \left. + \sigma_t \tilde{\mathbb{E}} \left[\zeta_{t,T}^{\frac{\alpha}{1-\alpha}} \int_t^T D_t(\sigma_s^{-1} B \mathcal{E}_{s,t}) d\tilde{W}_s \mid \xi_t, \mathcal{E}_t \right] \right\}, \quad t \in [0, T], \end{aligned}$$

where $\zeta_{t,s}^{-1} = \zeta_s^{-1}/\zeta_t^{-1}$, $\mathcal{E}_{t,s} = \mathcal{E}_s/\zeta_t^{-1}$, $\hat{X}_t = x_0 \tilde{E}[\zeta_T^{\frac{1}{\alpha-1}} | \xi_t, \mathcal{E}_t] / \tilde{E}[\zeta_t^{\frac{1}{\alpha-1}}]$.

In Corollary 1, (\mathcal{E}_t, ξ_t) is a sufficient statistic for the calculation of $\hat{\pi}_t$, and the Markovian nature of this statistic is very helpful for Monte Carlo simulations which we need to compute the second term of $\hat{\pi}_t$. Note that no third term appears like it does for stochastic interest rates.

The volatility model in Sect. 2 guarantees a complete market and allows for level dependency $\sigma_t = \sigma(S_t)$ as well as for dependencies on other \mathcal{F}^S -adapted state-variables like $\int_0^t S_s^i ds$, η_t , R_t . Using σ_t as one component of ξ_t , we can also implement dynamical models with respect to \tilde{W} . Since these often lead to unbounded processes we may have to consider $\sigma_t = h(\xi_t)$ for a bounded function h with suitably bounded and smooth derivatives. For empirical findings, surveys and connections to GARCH models see [2],[4].

Volatility models with dynamics w.r.t. a second Brownian motion which is not perfectly correlated to \tilde{W} (incomplete market) could be implemented using the methods in [8], but only under their strong assumptions.

But our model already covers a lot of empirical findings, like level dependency, imperfectly correlated stock-volatility, volatility clustering, fat tails, etc.. And since volatility results from the action of the traders it is reasonable that it depends on observable quantities. If we think of Y_t as an indicator of the current state of the economy, a model $\sigma_t = \sigma(\eta_t)$ that depends on the best estimate η_t for Y_t seems to be appropriate, e.g.:

Example 1. In the case $n = 1, d = 2$ (one stock, two states) we consider

$$\sigma_t = \sigma(\eta_t^1), \quad \text{where} \quad \sigma(x) = s_0 + s_1 x + s_2 x^2$$

with $s_0, -s_1, s_2, s_0 - s_1^2/(4s_2) > 0$. Since $\eta_t^1 \in (0, 1)$ we can extend σ outside $\overline{\sigma((0, 1))}$ such that the conditions in Sect. 2 are satisfied. The Malliavin derivative of $\xi = \eta^1$ needed in Theorem 2 is $D_t \eta_u^1 = -\zeta_u \eta_u^1 D_t \zeta_u^{-1} + \zeta_u D_t \mathcal{E}_u^1$.

4 Parameter Estimation and Approximation

For known covariance matrices the parameters of the drift process, Q and B , can be estimated using the EM-algorithm, cf. [1]. This can be done by considering the *occupation time* $O_t^k = \int_0^t Y_s^k ds$ in state e_k , the *number of jumps* $N^{kl} = \int_0^t Y_{s-}^k dY_s^l$ from e_k to e_l , $k \neq l$, and the *level integral* $G^k = \int_0^t Y_s^k dR_s$. Thinking of O^k, G^k, N^{kl} as observations under P parameterized by Q and B , we want to know under which measure P' , given by new parameters Q' and B' , our observation would have been the most probable. After an observation interval $[0, t]$, maximizing the likelihood function $E[\log(L_t) | \mathcal{F}_t^S]$, where $dP' = L_t dP$ on \mathcal{F}_t , leads to

$$Q'_{kl} = \mathcal{E}_t(N^{kl})/\mathcal{E}_t(O^k) \quad \text{and} \quad B'_k = \mathcal{E}_t(G^k)/\mathcal{E}_t(O^k) \quad (6)$$

for $k, l = 1, \dots, d$, $l \neq k$, where $\mathcal{E}_t(X) = \tilde{\mathbb{E}}[Z_t^{-1}X_t | \mathcal{F}_t^S]$ is the unnormalized filter of the \mathcal{F} -adapted process $X = (X_t)_{t \in [0, T]}$ (note that $\mathcal{E}_t(Y) = \mathcal{E}_t$).

Iterating this procedure we obtain a sequence of parameters, whose likelihood functions with respect to the initial measure are increasing and hence converge. Concavity implies that they converge to the maximum.

The Euler scheme applied to versions of the unnormalized filters in (6), which lead to robust versions of the corresponding filters, see [5], yields for time steps $\Delta t = t_k - t_{k-1}$ the approximations \mathcal{E}_k of \mathcal{E}_{t_k} etc.. In the corresponding equations in [9, Sect. 5] only “ $\sum_{j=1}^n (\sigma\sigma^\top)_{ij}^{-1} \Delta \tilde{R}_k^j$ ” has to be replaced by “ ΔR_k^i ”. As approximation for the filter we use $\eta_k = \mathcal{E}_k / \mathbf{1}_d^\top \mathcal{E}_k$.

Due to the use of Girsanov’s Theorem in its derivation the continuous time EM algorithm cannot be extended to estimate the covariance matrix. So it has to be estimated before. In [9] an algorithm is presented for estimating a constant covariance matrix very accurately. For the simple volatility model of Example 1 we propose the following ad-hoc algorithm, which leads to good results for the coefficients s_0, s_1, s_2 , cf. Sect. 5. Due to the dependence of the volatility on the filter which is based on the estimates for B, Q , the volatility would change in each step of the EM-algorithm. Therefore we estimate s_0, s_1, s_2 separately based on the prior estimates for B, Q and repeat the EM-algorithm for a fixed volatility process. The algorithm in detail:

- 1) Estimate the constant volatility using the method proposed in [9, Sect. 5].
- 2) Run the EM-algorithm once to find estimates for B, Q .
- 3) Compute the approximations η_k , $k = 1, \dots, N$ for the filter. Sort these and split them into m sets (of possibly different sizes). Compute the averages $\bar{\eta}_i$ of the filter values and the standard deviations $\bar{\sigma}_i$ of the corresponding returns (divided by $\Delta t^{1/2}$) in each set ($i = 1, \dots, m$). Perform a quadratic least square fit for $(\bar{\eta}_i, \bar{\sigma}_i)_{i=1, \dots, m}$ to find the parameters s_0, s_1, s_2 . Repeat this step n_1 times.
- 4) Compute the volatility process based on the estimates in 3).
- 5) Repeat the EM-algorithm n_2 times using the fixed volatility process in 4).
- 6) Repeat 3), ..., 5) n_3 times.

5 Simulated and Historical Prices

The following simulation and application to market data is meant to see if the simple volatility model of Example 1 can improve the results obtained assuming constant volatility. For more complex simulations (power utility, two stocks, but constant volatility) we refer to [9, Sect. 6]. We consider the simple model of one stock with only two states $b = (b_1, b_2)^\top$ for μ and use $U = \log$. Extending Corollary 1 to \mathcal{F}^S -adapted interest rates $(r_t)_{t \in [0, T]}$ yields

$$\hat{\pi}_t = \hat{X}_t(b^\top \eta_t - r_t) / \sigma_t^2, \quad t \in [0, T].$$

Simulation: We simulate stock prices for 6 years, each consisting of 252 trading days and based on the parameter estimation over the first 5 years we

apply different optimal strategies in the final year. We use the algorithm presented in Sect. 4 with $m = 23$, set sizes 10,10,10,25,25,50,50,100,..., 100,50,50,25,25,10,10,10, and $n_1 = 10, n_2 = 1, n_3 = 1$. For initial parameters $b_1 = \bar{\mu} + 0.5$, $b_2 = \bar{\mu} - 0.5$, where $\bar{\mu}\Delta t$ is the average return in the estimation interval, $\lambda_1 = 18$, $\lambda_2 = 12$, and 500 simulations we obtained the estimates in Table 1. We observe that the estimates for the volatility and the states are very good or reasonable but for the rates they are poor (close to initial values). For a discussion of comparable estimation results see [9].

Table 1. Parameter Estimation for Simulated Prices

parameters	b_1	b_2	λ_1	λ_2	s_0	s_1	s_2
true parameters:	0.80	-0.30	15	10	1.0	-3.4	3.2
estimated par.:	0.70	-0.28	17.0	12.7	1.11	-3.48	3.29
standard dev.:	0.17	0.20	0.77	0.44	0.27	1.15	1.22
abs. error in %:	13.0	7.27	13.5	26.5	10.6	2.26	2.83

For the optimization with initial capital $x_0 = 1$ and constant interest rate 0.06 we compare the optimal strategy with the buy-and-hold strategy (b/h) of investing in only the stock. If the wealth process becomes negative in the optimization interval, we abort the evaluation and set the utility equal -1 . The results are in Table 2. Since the averages (av.) are influenced by extreme events we also included the medians (med.). Bankruptcy occurred 3 times. The values using the true parameters are given in parentheses.

Table 2. Optimal Terminal Wealth for Simulated Prices

strategy	av. \hat{X}_T	med. \hat{X}_T	av. $\log(\hat{X}_T)$	med. $\log(\hat{X}_T)$
optimal	2.29 (6.81)	1.13 (1.24)	0.19 (0.37)	0.12 (0.21)
b/h	1.16	1.14	0.11	0.13

Historical prices: We consider 20 stocks of the Dow Jones Industrial Index and use daily prices for 30 years, 1972–2001, each with 252 trading days. We consider the corresponding historic interest rates (fed rates) as well. The setup is the same as for the simulated data, only that we now use $\lambda_1 = \lambda_2 = 126$ as initial values. For each stock the procedure (5 years estimation, 1 year optimization) is carried out with starting years 1972, 1973,..., 1996. So we perform 500 experiments whose outcomes we average. For the optimization results in Table 3 we use as benchmarks also the Merton strategy using constant σ

and μ , and the optimal HMM strategy assuming constant σ (const. σ). The parameter estimation yields on average $s_0 \approx 1.19$, $s_1 \approx -3.66$, $s_2 \approx 3.52$, $b_1 \approx 0.70$, $b_2 \approx -0.39$, and $\lambda_1 = \lambda_2 \approx 125.6$.

Table 3. Optimal Terminal Wealth for Historical Prices

strategy	av. \hat{X}_T	med. \hat{X}_T	av. $\log(\hat{X}_T)$	med. $\log(\hat{X}_T)$	aborted
optimal	1.745	1.181	0.122	0.167	1
const. σ	1.603	1.138	0.057	0.129	11
Merton	1.163	1.095	0.006	0.091	2
b/h	1.153	1.121	0.116	0.114	—

Conclusion: The optimal strategy based on the new volatility model clearly improves the optimal strategy that assumes constant volatility. This might be because it assigns higher volatilities to high or low values of the filter (as it is observed in the data), and hence extreme long or short positions are avoided. Volatility models with the same property should also provide good results, but the parameter estimation can be more difficult.

References

1. Elliott, R. J. (1993): New finite-dimensional filters and smoothers for noisily observed Markov chains. *IEEE Trans. on Information Theory* 39 (1), 265–271
2. Ghysels, E., Harvey, A., Renault, E. (1996): Stochastic volatility. In: Maddala, G. S., Rao, C. R., Vinod, H. D. (Eds.) *Handbook of Statistics Vol. 14: Statistical Methods in Finance*. North-Holland, Amsterdam, 116–191.
3. Haussmann, U. G., Sass, J.: Optimal terminal wealth under partial information for HMM stock returns, *Proceedings of the AMS-IMS-SIAM Summer Conference on Mathematics of Finance*, Utah 2003, *AMS Contemporary Math.*, to appear
4. Hobson, D. G., Rogers, L. C. G. (1998): Complete models with stochastic volatility. *Mathematical Finance* 8, 27–48
5. James, M. R., Krishnamurthy, V., Le Gland, F. (1996): Time discretization of continuous-time filters and smoothers for HMM parameter estimation. *IEEE Transactions on Information Theory* 42 (2), 593–605
6. Karatzas, I., Ocone, D. L., Li, J. (1991): An extension of Clark’s formula. *Stochastics and Stochastics Reports* 37, 127–131
7. Lakner, P. (1998): Optimal trading strategy for an investor: the case of partial information. *Stochastic Processes and their Applications* 76, 77–97
8. Pham, H., Quenez, M.-C. (2001): Optimal portfolio in partially observed stochastic volatility models. *The Annals of Applied Probability* 11/1, 210 – 238
9. Sass, J., Haussmann, U. G. (2003): Optimizing the terminal wealth under partial information: The drift process as a continuous time Markov chain. Preprint, www.math.ubc.ca/~uhauss/

On the Set of Optimal Policies in Variance Penalized Markov Decision Chains

Karel Sladký and Milan Sitař

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 18208 Praha 8, Czech Republic
e-mail: sladky@utia.cas.cz

Abstract. In this note we present a policy iteration algorithm for constructing a set of efficient stationary policies containing optimal policies with respect to various criteria used for the mean variance tradeoff. This algorithm works both for the unichain and multichain models. We show that the obtained policies are optimal also in the class of Markovian (memoryless) policies.

1 Introduction and Preliminaries

The usual optimization criteria examined in the literature on stochastic dynamic programming, e.g. total (discounted) or mean reward, may be quite insufficient to characterize the problem from the point of the decision maker. To this end it may be necessary to select more sophisticated criteria that reflects also the variability-risk features of the problem, e.g. by taking into account also higher moments and variance of the cumulative rewards. For a detailed discussion of these approaches see the review paper by White [14].

Some successful research on criteria reflecting the variability-risk features by considering the mean-variance tradeoff has been reported in the literature on Markovian decision models. It is important to notice, except of the papers minimizing the variance in the class of mean optimal or sensitive optimal policies (see Jacquette [2], [3], Mandl [7]), the “variance” is usually considered only with respect to one-stage reward variances and not to the variance of cumulative rewards. In particular, Sobel [13] analyzed the problem of maximization the ratio of the mean to the standard deviation for undiscounted unichain model using the methods of non-linear and parametric linear programming. Under the same assumptions Kawai [6] considered the problem of minimizing the variance subject to a lower bound on the mean. Benito [1] presented a recursive formula for expected cumulative reward and its variance provided that one-stage rewards are random variables. Filar, Kallenberg, Huang and Lee [4], [5] proposed a mathematical programming approach to mean-variance Markov decision chains and unified and extended the previous results. Sladký and Sitař [11] considered connections between the variance of cumulative and one-stage rewards, and for the unichain models established a policy iteration method for finding the class efficient stationary policies

containing optimal policies with respect to any criterion common for the mean-variance tradeoff.

In this note we extend the approach used in [11] to the multichain models and show that policies contained in the class of efficient stationary policies are optimal also in the class of all Markovian (memoryless) policies.

We consider a classical controlled Markov chain $X = \{X_n, n = 0, 1, \dots\}$ with finite state space $\mathcal{S} = \{1, 2, \dots, s\}$, transition probabilities p_{ij}^k and immediate rewards r_{ij}^k depending on the selected decision k from a finite set D_i for every $i, j \in \mathcal{S}$. Obviously, $r_i^k = \sum_{j=1}^s p_{ij}^k r_{ij}^k$ is the expected one-stage reward, resp. $r_i^{(2),k} = \sum_{j=1}^s p_{ij}^k [r_{ij}^k]^2 \geq 0$ is the corresponding second moment, obtained in state $i \in \mathcal{S}$ if decision $k \in D_i$ is selected and $\sum_{j=1}^s p_{ij}^k [r_{ij}^k - r_i^k]^2 = r_i^{(2),k} - [r_i^k]^2$ is the corresponding one-stage reward variance. Let $D = D_1 \times D_2 \times \dots \times D_s$. We assume that the initial distribution of the chain $\alpha = [\alpha_1, \dots, \alpha_s]$ is given.

We restrict attention on Markovian (memoryless) policies Π controlling the chain, i.e. $\Pi = (\pi^0, \pi^1, \pi^2, \dots)$ where $\pi^n \in D$ for every $n = 0, 1, 2, \dots$ and $\pi_i^n \in D_i$ is the decision at the n th transition when the chain is in state i . A policy which takes at all times the same decision rule is called stationary and is identified by $\Pi \sim (\pi)$. Let $\Pi^{(1)} \sim (\pi^{(1)})$, $\Pi^{(2)} \sim (\pi^{(2)})$ be two stationary policies. Then stationary policy $\Pi \sim (\pi)$ selecting in state i decision $\pi_i^{(1)}$ (resp. decision $\pi_i^{(2)}$) with probability p_i (resp. $1 - p_i$) is a policy arising by randomization of policies $\Pi^{(1)} \sim (\pi^{(1)})$ and $\Pi^{(2)} \sim (\pi^{(2)})$.

For what follows it will be convenient to use more condensed matrix notations. We denote by $P(\pi^n)$ be the $s \times s$ matrix whose ij th element equals $p_{ij}^{\pi_i^n}$ and let $P^m(\Pi) = \prod_{n=0}^{m-1} P(\pi^n)$ (obviously, $P^{n+1}(\Pi) = P^n(\Pi)P(\pi^n)$, for convenience we set $P^0(\Pi) = I$, the identity matrix). If $\Pi \sim (\pi)$ (i.e. if Π is stationary) then $P^m(\Pi) = [P(\pi)]^m$. Recall that the limiting matrix $P^*(\pi) = \lim_{m \rightarrow \infty} m^{-1} \sum_{n=0}^{m-1} [P(\pi)]^n$ exists. In particular, if $P(\pi)$ is *unichain* (i.e. $P(\pi)$ contains a single class of recurrent states) the rows of $P^*(\pi)$, denoted $p^*(\pi)$, are identical. Similarly, $r(\pi^n) = r^{(1)}(\pi^n)$, resp. $r^{(2)}(\pi^n)$, denotes the row vector whose i th element equals $r_i^{\pi_i^n}$, resp. $r_i^{(2), \pi_i^n}$.

If we denote by $v_i^{(1),n}(\Pi)$ total expected reward earned in the n next transitions provided the chain starts in state i and policy Π is followed, then for the (column) vector of total expected rewards $v^{(1),n}(\Pi)$ (whose i th element is $v_i^{(1),n}(\Pi)$) we have $v^{(1),n}(\Pi) = \sum_{k=0}^{n-1} P^k(\Pi) r(\pi^k)$. Moreover, $g^{(1)}(\Pi) := \lim_{n \rightarrow \infty} n^{-1} v^{(1),n}(\Pi)$ (provided the limit exists) is the vector of the mean rewards, i.e. of the long run average expected rewards. Observe that the i th element of $g^{(1)}(\Pi)$, denoted by $g_i^{(1)}(\Pi)$, is the mean reward if the Markov chain starts in state i . In particular, for stationary policy $\Pi \sim (\pi)$ we get $v^{(1),n}(\Pi) = \sum_{k=0}^{n-1} [P(\pi)]^k r(\pi)$ and for n tending to infinity we have $g^{(1)}(\Pi) = \lim_{n \rightarrow \infty} n^{-1} \sum_{k=0}^{n-1} [P(\pi)]^k r(\pi) = P^*(\pi) r(\pi)$. If the initial state distribution is equal to $\alpha = [\alpha_1, \dots, \alpha_s]$ the total expected reward and

the mean reward is given by $\bar{v}^{(1),n}(\Pi) = \sum_{j=1}^s \alpha_j v_j^{(1),n}(\Pi)$ and by $\bar{g}^{(1)}(\Pi) = \sum_{j=1}^s \alpha_j g_j^{(1)}(\Pi)$ (or by $\bar{v}^{(1),n}(\Pi) = \alpha \cdot v_j^{(1),n}(\Pi)$ and by $\bar{g}^{(1)}(\Pi) = \alpha \cdot g^{(1)}(\Pi)$, where \cdot denotes the inner product) respectively. In case that $P(\pi)$ is unichain, the rows of $P^*(\pi)$ are identical, equal to $p^*(\pi)$, and $g^{(1)}(\pi)$ is a constant vector independent of the initial state distribution α .

Policy $\hat{\Pi}^{(1)}$ is called (first moment) average optimal (or mean optimal) if

$$\liminf_{m \rightarrow \infty} m^{-1} v^m(\hat{\Pi}^{(1)}) \geq \liminf_{m \rightarrow \infty} m^{-1} v^m(\Pi) \quad \text{for every policy } \Pi. \quad (1)$$

Similarly for the second moments $r_i^{(2),k}$ of one-stage rewards earned in state i under decision $k \in D_i$ we have $v^{(2),n}(\Pi) := \sum_{k=0}^{n-1} P^k(\Pi) r^{(2)}(\pi^k)$ with $g^{(2)}(\Pi) := \lim_{n \rightarrow \infty} n^{-1} v^{(2),n}(\Pi)$ (provided the limit exists); for stationary policy $\Pi \sim (\pi)$ we have $g^{(2)}(\Pi) = \lim_{n \rightarrow \infty} n^{-1} \sum_{k=0}^{n-1} [P(\pi)]^k r^{(2)}(\pi) = P^*(\pi) r^{(2)}(\pi)$. In case that $P(\pi)$ is unichain $g^{(2)}(\Pi)$ is a constant vector.

Policy $\hat{\Pi}^{(2)}$ is second moment average minimal if for every policy $\Pi = (\pi^n)$

$$\limsup_{m \rightarrow \infty} m^{-1} v^{(2),m}(\hat{\Pi}^{(2)}) \leq \limsup_{m \rightarrow \infty} m^{-1} v^{(2),m}(\Pi) \quad (2)$$

It is well known from the literature (see e.g. [8], [9]) that the average optimal policy, and hence also the second moment average minimal policy, can be found in the class of stationary policies, i.e. there exists $\hat{\Pi}^{(\ell)} \sim (\hat{\pi}^{(\ell)})$ (for $\ell = 1, 2$) such that (1), (2) is fulfilled where $g^{(\ell)}(\hat{\pi}^{(\ell)})$, $w^{(\ell)}(\hat{\pi}^{(\ell)})$ is the unique solution to the set of equations

$$\max_{\pi \in \bar{D}} \psi^{(1)}(\pi, \hat{\pi}^{(1)}) = 0 \quad \text{with} \quad \psi^{(1)}(\pi, \hat{\pi}) \stackrel{\text{def}}{=} [P(\pi) - I] g^{(1)}(\hat{\pi}) \quad (3)$$

$$\max_{\pi \in \bar{D}} \varphi^{(1)}(\pi, \hat{\pi}^{(1)}) = 0 \quad \text{with} \quad \varphi^{(1)}(\pi, \hat{\pi}) \stackrel{\text{def}}{=} r(\pi) - g^{(1)}(\hat{\pi}) + [P(\pi) - I] w^{(1)}(\hat{\pi}) \quad (4)$$

where $\bar{D} = \{\pi \in D : \max_{\pi \in \bar{D}} \psi^{(1)}(\pi, \hat{\pi}^{(1)}) = 0\}$ and $P^*(\hat{\pi}^{(1)}) w^{(1)}(\hat{\pi}^{(1)}) = 0$.

Analogous formulas (with max replaced by min) hold for the $\ell = 2$ and the second moment average minimal policies. The symbol $\psi_i^{(\ell)}(\cdot, \cdot)$, $\varphi_i^{(\ell)}(\cdot, \cdot)$ denotes the i th entry of $\psi^{(\ell)}(\cdot, \cdot)$, $\varphi^{(\ell)}(\cdot, \cdot)$.

After some algebra we can show that in virtue of (3), (4) (cf. ([10])) for any $\Pi = (\pi^n)$, $\hat{\Pi} \sim (\hat{\pi})$, $\ell = 1, 2$, and $n = 0, 1, \dots$

$$\begin{aligned} v^{(\ell),n}(\Pi) &= n g^{(\ell)}(\hat{\pi}) + [I - P^n(\Pi)] w^{(\ell)}(\hat{\pi}) \\ &+ \sum_{k=0}^{n-1} P^k(\Pi) [(n-1-k) \psi^{(\ell)}(\pi^k, \hat{\pi}) + \varphi^{(\ell)}(\pi^k, \hat{\pi})]. \end{aligned} \quad (5)$$

On premultiplying (5) by the initial state distribution $\alpha = [\alpha_1, \dots, \alpha_s]$ we have for (scalar values) $\bar{v}^{(\ell),n}(\Pi) = \alpha \cdot v^{(\ell),n}(\Pi)$, $\bar{g}^{(\ell)}(\hat{\pi}) = \alpha \cdot g^{(\ell)}(\hat{\pi})$, $\bar{w}^{(\ell)}(\hat{\pi}) = \alpha \cdot w^{(\ell)}(\hat{\pi})$ and $p^n(\Pi) = \alpha \cdot P^n(\Pi)$

$$\begin{aligned} \bar{v}^{(\ell),n}(\Pi) &= n \bar{g}^{(\ell)}(\hat{\pi}) + \bar{w}^{(\ell)}(\hat{\pi}) - p^n(\Pi) \bar{w}^{(\ell)}(\hat{\pi}) \\ &+ \sum_{k=0}^{n-1} p^k(\Pi) [(n-1-k) \bar{\psi}^{(\ell)}(\pi^k, \hat{\pi}) + \bar{\varphi}^{(\ell)}(\pi^k, \hat{\pi})]. \end{aligned} \quad (6)$$

Moreover, for stationary $\Pi \sim (\pi)$ from (5) we conclude that

$$g^{(\ell)}(\pi) - g^{(\ell)}(\tilde{\pi}) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} P^k(\Pi) [(n-1-k)\psi^{(\ell)}(\pi, \tilde{\pi}) + \varphi^{(\ell)}(\pi, \tilde{\pi})]. \quad (7)$$

Since the set of all stationary policies is *finite*, $n\psi^{(\ell)}(\pi, \tilde{\pi}) + \varphi^{(\ell)}(\pi, \tilde{\pi})$ is uniformly bounded in $\pi \in D$ and n if and only if $\psi^{(\ell)}(\pi, \tilde{\pi}) = 0$. Hence from (7) it follows that if $\psi_i^{(\ell)}(\pi, \tilde{\pi}) \neq 0$ the state i must be transient under policy $\Pi \sim (\pi)$.

Supposing that policy $\Pi = (\pi^n)$ is followed, it is reasonable to consider in state i the one-stage variance as

$$\sum_{j=1}^s p_{ij}^{\pi_i^n} [r_{ij}^{\pi_i^n} - g_i^{(1)}(\Pi)]^2 = \sum_{j=1}^s p_{ij}^{\pi_i^n} \{[r_{ij}^{\pi_i^n}]^2 - 2g_i^{(1)}(\Pi)r_{ij}^{\pi_i^n} + [g_i^{(1)}(\Pi)]^2\} = r_i^{(2), \pi_i^n} - 2g_i^{(1)}(\Pi)r_i^{\pi_i^n} + [g_i^{(1)}(\Pi)]^2.$$

In case that $\Pi \sim (\pi)$ on denoting by $G(\Pi) = \text{diag}\{g_i^{(1)}(\Pi)\}$ we conclude that

$$\begin{aligned} y^n(\Pi) &\stackrel{\text{def}}{=} \sum_{k=0}^{n-1} P^k(\pi) \{r^{(2)}(\pi) - 2G(\Pi)r^{(1)}(\pi) + G(\Pi)g^{(1)}(\Pi)\} \\ &= v^{(2),n}(\Pi) - 2G(\Pi)v^{(1),n}(\Pi) + nG(\Pi)g^{(1)}(\Pi). \end{aligned} \quad (8)$$

Hence the mean (one-stage) reward variance $\sigma^2(\Pi) = \lim_{n \rightarrow \infty} n^{-1}y^n(\Pi)$ and from (8) we conclude that for stationary policy $\Pi \sim (\pi)$

$$\sigma^2(\Pi) = P^*(\pi)r^{(2)}(\pi) - [P^*(\pi)r^{(1)}(\pi)]_{\text{sq}} = g^{(2)}(\Pi) - [g^{(1)}(\Pi)]_{\text{sq}} \quad (9)$$

($[r(\cdot)]_{\text{sq}}$ results from $r(\cdot)$ by squaring each entry). In particular, for the i th element of $\sigma^2(\Pi)$ we have $\sigma_i^2(\Pi) = g_i^{(2)}(\Pi) - [g_i^{(1)}(\Pi)]^2$ and if the initial distribution of the chain is given by $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_s]$ then the corresponding mean reward variance $\bar{\sigma}^2(\Pi) = \sum_{i=1}^s \alpha_i \{g_i^{(2)}(\Pi) - [g_i^{(1)}(\Pi)]^2\} = \alpha \cdot g^{(2)}(\Pi) - \alpha \cdot [g^{(1)}(\Pi)]_{\text{sq}}$.

2 Mean-Variance Tradeoff

On the base of the first two moments of one-stage rewards we can formulate the following optimality criteria for mean-variance tradeoff in undiscounted Markov decision chains. Observe that by (9) for policy $\Pi = (\pi^n)$ we have $\sigma_i^2(\Pi)/g_i^{(1)}(\Pi) = g_i^{(2)}(\Pi)/g_i^{(1)}(\Pi) - g_i^{(1)}(\Pi)$ and that $\sigma_i^2(\Pi)/[g_i^{(1)}(\Pi)]^2 = g_i^{(2)}(\Pi)/[g_i^{(1)}(\Pi)]^2 - 1$, provided that $g_i^{(1)}(\Pi)$, $g_i^{(2)}(\Pi)$ exist.

i) *Weighted mean variance optimality.* Considering this criterion, policy $\hat{\Pi}^{(\delta)}$ is called δ -weighted mean variance optimal if for every policy $\Pi = (\pi^n)$ and every $i \in \mathcal{S}$ (with $\delta \in [0, 1]$ fixed)

$$h_i^{(\delta)}(\Pi) = (1-\delta) \frac{g_i^{(2)}(\Pi)}{g_i^{(1)}(\Pi)} - \delta g_i^{(1)}(\Pi) \geq (1-\delta) \frac{g_i^{(2)}(\hat{\Pi}^{(\delta)})}{g_i^{(1)}(\hat{\Pi}^{(\delta)})} - \delta g_i^{(1)}(\hat{\Pi}^{(\delta)}) \quad (10)$$

Moreover, if the initial distribution of the chain is given by $\alpha = [\alpha_1, \dots, \alpha_s]$ then from (10) it follows that $\bar{h}^{(\delta)}(\Pi) = \sum_{j=1}^n \alpha_j h_j^{(\delta)}(\Pi) \geq \bar{h}^{(\delta)}(\hat{\Pi}^{(\delta)})$.

From (10) we immediately conclude that:

If $\delta = 1$ then policy $\hat{\Pi}^{(\delta)}$ is (first moment) average optimal (cf. (1));

If $\delta = 0$ then policy $\hat{\Pi}^{(\delta)}$ minimizes the ratio of the average second moment of one-stage rewards to the mean reward;

If $\delta = \frac{1}{2}$ then policy $\hat{\Pi}^{(\delta)} \equiv \hat{\Pi}$ is mean variance optimal, i.e. for every policy $\Pi = (\pi^n)$ and every $i \in \mathcal{S}$

$$h_i(\Pi) = \frac{g_i^{(2)}(\Pi)}{g_i^{(1)}(\Pi)} - g_i^{(1)}(\Pi) \geq \frac{g_i^{(2)}(\hat{\Pi})}{g_i^{(1)}(\hat{\Pi})} - g_i^{(1)}(\hat{\Pi}). \quad (11)$$

ii) *Square mean variance optimality.* Policy Π^* is called *square mean variance optimal* if

$$h_i^{(2)}(\Pi) = \frac{g_i^{(2)}(\Pi)}{[g_i^{(1)}(\Pi)]^2} \geq \frac{g_i^{(2)}(\Pi^*)}{[g_i^{(1)}(\Pi^*)]^2} \quad (12)$$

for every policy $\Pi = (\pi^n)$ and every $i \in \mathcal{S}$ implying that for a given initial distribution $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_s]$ $\bar{h}^{(2)}(\Pi) = \sum_{j=1}^n \alpha_j h_j^{(2)}(\Pi) \geq \bar{h}^{(2)}(\Pi^*)$.

iii) *Weighted square mean variance optimality.* Policy $\Pi^{*,\delta}$ is called δ -*weighted square mean variance optimal* if for every policy $\Pi = (\pi^n)$ and every $i \in \mathcal{S}$

$$\begin{aligned} h_i^{(2,\delta)}(\Pi) &= \delta \{g_i^{(2)}(\Pi) - [g_i^{(1)}(\Pi)]^2\} - (1 - \delta)[g_i^{(1)}(\Pi)]^2 \geq \\ &\geq \delta \{g_i^{(2)}(\Pi^{*,\delta}) - [g_i^{(1)}(\Pi^{*,\delta})]^2\} - (1 - \delta)[g_i^{(1)}(\Pi^{*,\delta})]^2 \end{aligned} \quad (13)$$

that can be also written as

$$\delta g_i^{(2)}(\Pi) - [g_i^{(1)}(\Pi)]^2 \geq \delta g_i^{(2)}(\Pi^{*,\delta}) - [g_i^{(1)}(\Pi^{*,\delta})]^2. \quad (14)$$

Observe that if $\delta = 0$ we are looking for a mean optimal policy, if $\delta = 1$ we are looking for a policy minimizing the mean (one-stage) reward variance.

We can easily obtain some rough bounds on the objective functions for the (weighted) mean variance and (weighted) square mean variance optimality. In particular, (10), (12) and (14) respectively is bounded from below by

$$\begin{aligned} (1 - \delta) g_i^{(2)}(\hat{\pi}^{(0)})/g_i^{(1)}(\hat{\pi}^{(0)}) - \delta g_i^{(1)}(\hat{\pi}^{(1)}), \quad g_i^{(2)}(\hat{\pi}^{(0)})/g_i^{(1)}(\hat{\pi}^{(0)}) g_i^{(1)}(\hat{\pi}^{(1)}) \\ \text{and by } \delta g_i^{(2)}(\hat{\pi}^{(0)}) - [g_i^{(1)}(\hat{\pi}^{(1)})]^2 \quad \text{respectively.} \end{aligned}$$

Here $\hat{\Pi}^{(0)} \sim (\hat{\pi}^0)$ is reserved for stationary policy minimizing the ratio $g_i^{(2)}(\Pi)/g_i^{(1)}(\Pi)$, i.e. we are looking for optimal policies of a semi-Markov decision process for which if decision $k \in D_i$ is selected in state $i \in \mathcal{S}$ we obtain immediate return $r_i^{(2),k}$ and the corresponding sojourn time in state $i \in \mathcal{S}$ is equal to r_i^k (for details see e.g. [8], [9]). To obtain finer results we must employ more sophisticated analysis.

2.1 Unichain Models

Assumption 1. $P(\pi)$ contains a single class of recurrent states for any $\pi \in D$. Observe that then $\bar{g}^{(1)}(\Pi)$, $\bar{g}^{(2)}(\Pi)$ are constant vectors with elements $\bar{g}^{(1)}(\Pi)$, $\bar{g}^{(2)}(\Pi)$ independent of the initial distribution α .

In virtue of Assumption 1, for any (nonstationary, Markovian) policy $\Pi = (\pi^n)$ and any (stationary) policy $\tilde{\Pi} \sim (\tilde{\pi})$ from (5) we have for $\ell = 1, 2$

$$v^{(\ell),n}(\Pi) = ng^{(\ell)}(\tilde{\pi}) + [I - P^n(\Pi)]w^{(\ell)}(\tilde{\pi}) + \sum_{k=0}^{n-1} P^k(\Pi)\varphi^{(\ell)}(\pi^k, \tilde{\pi}) \quad (15)$$

For stationary $\Pi \sim (\pi)$ after some algebra we obtain from (15)

$$a(\tilde{\pi}) := \frac{\bar{g}^{(2)}(\Pi) - \bar{g}^{(2)}(\tilde{\Pi})}{\bar{g}^{(1)}(\Pi) - \bar{g}^{(1)}(\tilde{\Pi})} = \frac{p^*(\pi) \cdot \varphi^{(2)}(\pi, \tilde{\pi})}{p^*(\pi) \cdot \varphi^{(1)}(\pi, \tilde{\pi})} \leq \max_{i \in \mathcal{S}} \frac{\varphi_i^{(2)}(\pi, \tilde{\pi})}{\varphi_i^{(1)}(\pi, \tilde{\pi})} \quad (16)$$

being the main ingredient to the following important result depicted also in Figure 1. (Further details along with an algorithmic procedure for constructing vertices of the convex hull $\bar{\mathcal{P}}$ can be found in [11]).

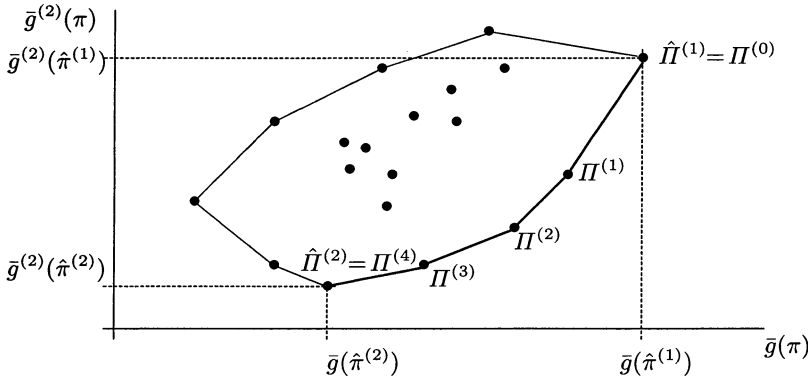


Fig. 1. Convex hull of the set of stationary policies.

Proposition 1. Consider in \mathbb{R}^2 the set \mathcal{P} of pairs $(\bar{g}^{(1)}(\Pi), \bar{g}^{(2)}(\Pi))$ for all (pure) stationary policies $\Pi \sim (\pi)$ with $\pi \in D$ along with the corresponding convex hull $\bar{\mathcal{P}}$. Then the adjacent vertices of $\bar{\mathcal{P}}$ are determined by (stationary) policies $\pi \in D$ that differ in the decision only for one state. In particular, for the pairs $(\bar{g}^{(1)}(\Pi^{(k)}), \bar{g}^{(2)}(\Pi^{(k)}))$, $k = 0, \dots, K$, on the south-east boundary of $\bar{\mathcal{P}}$ we have

$$\bar{a}(\pi^{(k)}) := \frac{\bar{g}^{(2)}(\Pi^{(k+1)}) - \bar{g}^{(2)}(\Pi^{(k)})}{\bar{g}^{(1)}(\Pi^{(k+1)}) - \bar{g}^{(1)}(\Pi^{(k)})} = \max_{i \in \mathcal{S}} \max_{\substack{\pi \in D \\ \varphi_i^{(1)}(\pi, \pi^{(k)}) < 0}} \frac{\varphi_i^{(2)}(\pi, \pi^{(k)})}{\varphi_i^{(1)}(\pi, \pi^{(k)})} \quad (17)$$

Theorem 1. Let $\Pi = (\pi^n)$ be a nonstationary (Markovian) policy. Then:

- (i) For $g^{(1)}(\Pi)$, $g^{(2)}(\Pi)$ we have $(\bar{g}^{(1)}(\Pi), \bar{g}^{(2)}(\Pi)) \in \bar{\mathcal{P}}$.
- (ii) Moreover, (weighted) mean variance optimal and also (weighted) square

mean variance optimal policies can be selected in the class of stationary policies $\Pi^{(k)} \sim (\pi^{(k)})$ ($k = 0, \dots, K$) on the south-east boundary of $\bar{\mathcal{P}}$.

Proof. To verify (i), since $\bar{g}^{(1)}(\Pi)$, resp. $\bar{g}^{(2)}(\Pi)$, is bounded from above, resp. from below, by $\bar{g}^{(1)}(\hat{\pi}^{(1)})$, resp. $\bar{g}^{(2)}(\hat{\pi}^{(2)})$, let us assume existence of $\Pi = (\pi^n)$ such that $\bar{g}^{(1)}(\Pi) \in [\bar{g}^{(1)}(\Pi^{(k+1)}), \bar{g}^{(1)}(\Pi^{(k)})]$ and $\bar{g}^{(2)}(\Pi) - \bar{g}^{(2)}(\Pi^{(k)}) < \bar{a}(\pi^{(k)})[\bar{g}^{(1)}(\Pi) - \bar{g}^{(1)}(\Pi^{(k)})]$ (see Figure 1). Mimicking the reasoning used in (15), (16), we then conclude existence of $i_0 \in \mathcal{S}$, $\pi \in D$ such that $\varphi_{i_0}^{(1)}(\pi, \pi^{(k)}) < 0$ and $\varphi_{i_0}^{(2)}(\pi, \pi^{(k)}) < \bar{a}(\pi^{(k)})\varphi_{i_0}^{(1)}(\pi, \pi^{(k)})$ that contradicts (17). Hence we can restrict our attention on the class of stationary possibly randomized policies. As it was shown in [11] assertion (ii) is fulfilled in the class of stationary policies.

2.2 Multichain Models

Similarly, as in the unichain case, we find (e.g. by the policy iteration algorithm) stationary policy $\hat{\Pi}^{(1)} \sim (\hat{\pi}^{(1)})$ maximizing mean reward (observe that, in general, $\hat{\Pi}^{(1)} \sim (\hat{\pi}^{(1)})$ defines a Markov chain with several recurrent classes). Then, extending slightly the procedure used for unichain models, we construct a sequence of stationary policies (for $k = 0, 1, \dots, K$)

$$\hat{\Pi}^{(1)} \equiv \Pi^{(0)} \sim (\pi^{(0)}), \quad \Pi^{(1)} \sim (\pi^{(1)}), \quad \dots, \quad \Pi^{(k)} \sim (\pi^{(k)}), \quad \dots \quad (18)$$

$$\pi_i^{(k)} \neq \pi_i^{(k+1)} \text{ for one } i_k \in \mathcal{S}, \quad \pi_i^{(k)} = \pi_i^{(k+1)} \text{ for every } i \neq i_k \in \mathcal{S} \quad (19)$$

using the following iterative procedure:

Algorithm.

Step 0. Let $\hat{\Pi}^{(1)} \equiv \Pi^{(0)}$ be the mean optimal policy fulfilling conditions (3),(4).

Step 1. For $k = 0, 1, \dots$ calculate the values $a^{(k)}$ where

$$a^{(k)} := \max_{i \in \mathcal{S}} \left\{ \max \left[\max_{\substack{\pi \in D \\ \psi_i^{(1)}(\pi, \pi^{(k)}) < 0}} \frac{\psi_i^{(2)}(\pi, \pi^{(k)})}{\psi_i^{(1)}(\pi, \pi^{(k)})}; \max_{\substack{\pi \in D \\ \varphi_i^{(1)}(\pi, \pi^{(k)}) < 0}} \frac{\varphi_i^{(2)}(\pi, \pi^{(k)})}{\varphi_i^{(1)}(\pi, \pi^{(k)})} \right] \right\} \quad (20)$$

If $a^{(k)} > 0$ calculate the values $\bar{g}^{(1)}(\pi^{(k)})$ and $\bar{g}^{(2)}(\pi^{(k)})$ and repeat Step 1, else go to Step 2.

Step 2. For the considered optimality criterion calculate its values for all $\Pi^{(k)} \sim (\pi^{(k)})$ and select among them the optimal value.

To verify that the above procedure generates vertices on the south-east boundary of the convex hull $\bar{\mathcal{P}}$, first observe that on premultiplying (7) considered for $\ell = 1, 2$ by the initial state distribution α we conclude that

$$\begin{aligned} \frac{\bar{g}^{(2)}(\pi) - \bar{g}^{(2)}(\tilde{\pi})}{\bar{g}^{(1)}(\pi) - \bar{g}^{(1)}(\tilde{\pi})} &= \lim_{n \rightarrow \infty} \frac{\sum_{k=0}^{n-1} p^k(\Pi) [(n-1-k)\psi^{(2)}(\pi, \tilde{\pi}) + \varphi^{(2)}(\pi, \tilde{\pi})]}{\sum_{k=0}^{n-1} p^k(\Pi) [(n-1-k)\psi^{(1)}(\pi, \tilde{\pi}) + \varphi^{(1)}(\pi, \tilde{\pi})]} \\ &\leq \max \left[\max_{i \in \mathcal{S}} \frac{\psi_i^{(2)}(\pi, \tilde{\pi})}{\psi_i^{(1)}(\pi, \tilde{\pi})}; \max_{i \in \mathcal{S}} \frac{\varphi_i^{(2)}(\pi, \tilde{\pi})}{\varphi_i^{(1)}(\pi, \tilde{\pi})} \right] \end{aligned} \quad (21)$$

In particular, if $\tilde{\pi} = \hat{\pi}^{(1)}$ (where $\hat{\Pi}^{(1)} \sim (\hat{\pi}^{(1)})$ is a mean optimal mean policy) similarly as in the unichain case we conclude that for any policy $\Pi = (\pi^n)$

$$\frac{\{\bar{g}^{(2)}(\Pi) - \bar{g}^{(2)}(\hat{\pi}^{(1)})\}}{\{\bar{g}^{(1)}(\Pi) - \bar{g}^{(1)}(\hat{\pi}^{(1)})\}} \leq \max_{i \in \mathcal{S}} \left\{ \max \left[\max_{\substack{\pi \in D \\ \psi_i^{(1)}(\pi, \pi^{(k)}) < 0}} \frac{\psi_i^{(2)}(\pi, \pi^{(k)})}{\psi_i^{(1)}(\pi, \pi^{(k)})}; \max_{\substack{\pi \in D \\ \varphi_i^{(1)}(\pi, \pi^{(k)}) < 0}} \frac{\varphi_i^{(2)}(\pi, \pi^{(k)})}{\varphi_i^{(1)}(\pi, \pi^{(k)})} \right] \right\} \quad (22)$$

for $k = 0$. Repeating this reasoning we can construct the sequence of stationary policies $\Pi^{(k)} \sim (\pi^{(k)})$ ($k = 0, \dots, K$) being the vertices of the south-east boundary of convex hull of all stationary policies $\bar{\mathcal{P}}$. Similarly as in the unichain case these vertices are the set of efficient points with respect to the optimality criteria considered for the mean-variance tradeoff.

Acknowledgement. This research was supported by the Grant Agency of the Czech Republic under Grants 402/02/1015 and 402/01/0539.

References

1. Benito, F. (1982): Calculating the variance in Markov processes with random reward. *Trabajos de Estadística y de Investigación Operativa*, 33, 73–85
2. Jacquette, S.C. (1972): Markov decision processes with a new optimality criterion: Small interest rates. *Ann. Math. Statist.*, 43, 1894–1901
3. Jacquette, S.C. (1973): Markov decision processes with a new optimality criterion: Discrete time. *Ann. Statist.*, 1, 496–505
4. Filar, J., Kallenberg, L. C.M., Lee, H.-M. (1989): Variance penalized Markov decision processes. *Mathem. Oper. Research*, 14, 147–161
5. Ying Huang, Kallenberg, L. C.M. (1994): On finding optimal policies for Markov decision chains: a unifying framework for mean-variance-tradeoffs. *Mathem. Oper. Research*, 19, 434–448
6. Kawai, H. (1987): A variance minimization problem for a Markov decision process. *European J. Oper. Research*, 31, 140–145
7. Mandl, P. (1971): On the variance in controlled Markov chains. *Kybernetika*, 7, 1–12
8. Puterman, M.L. (1994): *Markov Decision Processes – Discrete Stochastic Dynamic Programming*. Wiley, New York
9. Ross, S.M. (1970): *Applied Probability Models with Optimization Applications*. Holden-Day, San Francisco, CA
10. Sladký, K. (1973): Necessary and sufficient optimality conditions for average rewards of controlled Markov chains. *Kybernetika*, 9, 124–137
11. Sladký, K., Sitař, M. (2004): Optimal solutions for undiscounted variance penalized Markov decision chains. In: *Dynamic Stochastic Optimization* (Marti, K., Ermoliev, Y., Pflug, G., Eds.), LNEMS, Vol. 532, Springer, Berlin, pp. 43–66
12. Sobel, M. J. (1982): The variance of discounted Markov decision processes. *J. Appl. Probab.*, 19, 794–802
13. Sobel, M. J. (1985): Maximal mean/standard deviation ratio in an undiscounted MDP. *Oper. Research Lett.*, 4, 157–159
14. White, D.J. (1988): Mean, variance and probability criteria in finite Markov decision processes: A review. *J. Optim. Theory Appl.*, 56, 1–29

On Random Sums and Compound Process Models in Financial Mathematics

Petr Volf

Institute of Information Theory and Automation AS CR,
182 08 Praha 8, Czech Republic,
E-mail: volf@utia.cas.cz

Abstract. We study the process composed from random increments occurring at random moments. Resulting compound process is therefore characterized by the intensity of random time points and by the distribution of increments. We propose a model considering the compound process as a two-dimensional random point process and expressing the mutual dependence of both components via the multiplicative hazard regression (Cox) model. The method of estimation of model components is presented and the prediction of process behaviour is studied. The application deals with the process of financial transactions and with the problem of detection of atypical trajectories.

1 Introduction, Compound Poisson Process

The compound Poisson process is the simplest case of the processes studied in the present paper. It consists of the homogeneous Poisson process of random time points, $0 < T_1 < T_2 < \dots$ with constant hazard rate h and of the increments Y_j occurring at times T_j and distributed exponentially with parameter g (i. e. $EY = 1/g$). Moreover, all components are mutually independent. The resulting process can be written as a random sum $C(t) = \sum Y_j \cdot 1[T_j \leq t]$, or formally also as

$$C(t) = \int_0^t Y(s) dN(s), \quad (C(0) = 0), \quad (1)$$

where $N(t)$ is the counting process corresponding to the Poisson process of time points.

In the following parts, we shall generalize the model, first to the non-homogeneous case, then to the case with dependent components. In most cases, the model of compound process is given by the hazard rate of random point process (or of connected counting process) and by the distribution of increments. In the present paper we propose a model characterizing both process parts with the help of hazard rates and also describing their mutual dependence with the aid of the multiplicative hazard regression model (Cox model). We shall recall the methods of estimation of model components and the properties of estimates, namely the consistency. Further, the methods

of the prediction of process behaviour will be studied. To this end, the results on crossing probabilities of risk processes (the ruin probabilities) will be utilized. The practical application will deal with the process of financial transactions and with the problem of detection of outlied (atypical) trajectories. The method uses the fact that the cumulated intensity actually represents also the transformation of the process to the scale of Poisson process with intensity one.

In certain cases it is necessary to distinguish between the hazard rate (or hazard function) and the intensity. By the hazard rate of a continuous random variable we mean $h(t) = f(t)/(1 - F(t))$, where $f(t)$, $F(t)$ are corresponding density and distribution function. More generally, hazard rate is a nonnegative function characterizing the random point process. The intensity then denotes the actual rate (local conditional probability) of random event occurrence, it can depend on actual development of the process and its covariates. Then, each trajectory of the process can have its own intensity (though they have the same model, the same hazard rate).

2 Crossing Probabilities for Compound Process

We shall now recall certain relevant results from the field of insurance mathematics. Let $C(t)$ represent a process of insurance claims, they should be covered from a fund $u + vt$ at time t . Then $R(t) = u + vt - C(t)$ is the risk process and the event $R(t) < 0$ means the ruin. Hence, the problem is how parameters u , v (initial capital and income rate) should be selected in order to keep the ruin probability $P(\inf_t R(t) < 0)$ less than given α (either on $[0, T]$ or $[0, \infty)$). Notice that the problem of selection of u , v is actually equivalent to the construction of (linear) prediction band, i.e. such a line that the process $C(t)$ lies below it with probability at least $1 - \alpha$. Though the basic results concerning the convolution formula for ruin probability, its Cramér–Lundberg approximation, etc., date back to 20-ties and 30-ties of the last century, the problem is solved explicitly only for the simplest cases. Namely, for the compound Poisson process and infinite time interval the exact solution reads:

$$P\left(\inf_t R(t) < 0\right) = \frac{h}{gv} \exp\left(-\frac{(v - h/g) \cdot u}{v/g}\right) = (1 - p) e^{-pug}, \quad (2)$$

where $p = (v - h/g)/v$ compares the growth rate of the reserve v with the mean of claims (per time unit) $h \cdot EY$, $p > 0$ is the basic condition for the existence of solution. Hence, for given α and selected $v > h/g$ corresponding $u = -\ln(\alpha/(1 - p))/(pg)$. For instance, for the compound Poisson process with $h = g = 1$, when we select $v = 1.4$, we compute $u = 9.31$, for faster slope $v = 1.6$ $u = 6.74$ is obtained, etc.

As regards the upper prediction limit for the trajectories of standard Poisson process (i.e. with hazard rate $h = 1$ and fixed increments equal to one),

they can be obtained (computed or randomly generated) from corresponding convolution formula. Namely, the probability of crossing the line $u + v \cdot t$ (for some $t > 0$, $u \geq 0, v > 1$) is given by

$$P(u) = (1 - p) \sum_{n=1}^{\infty} p^n \cdot \bar{F}^{(n)}(u),$$

where $p = 1/v$ and $1 - \bar{F}^{(n)}$ is the distribution function of the sum of n i.i.d. uniform on $(0, 1)$ random variables. Other possibility is to use the Cramér–Lundberg approximation $P(u) \sim (v-1)/(rv+1-v) \cdot \exp(-ru)$, where r is the positive solution to $\exp(r) - 1 = r \cdot v$. For instance, two of many possible 95% upper prediction bounds are given by $v = 1.2, u = 8.17$ and $v = 1.4, u = 4.35$.

There exists a number of papers and monographs dealing with the ruin probability problem, let us mention here [2], [3], [4]. At present the research is focused mainly to processes with sub-exponentially distributed increments (i.e. distributions with heavy tails) corresponding to many real situations. On the other hand, the models allowing for the dependence of increments and times (mutual or on common covariates and history) are not so frequent, due many both theoretical and methodological difficulties. We shall study one such case and propose at least an approximate method of solution.

3 Nonhomogeneous Compound Poisson Process

Let the process of times $N(t)$ be nonhomogeneous Poisson process with hazard rate $h(t)$ and the distribution of increments $Y(t)$ be given by hazard rate $g(y)$. We still assume that both components are independent mutually; this is the most serious restriction of such a model. Increments are therefore i.i.d. random variables, however, we describe them as a set of point processes $S_j(y)$, say, with only (maximally, in the case of censoring) one point – the value of increment. Denote $H(t) = \int_0^t h(s) ds$ and $G(y) = \int_0^y g(x) dx$ cumulated hazard rates.

Let n realizations be observed in $[0, T]$ so that the data are $N_i(t)$, $C_i(t)$, $i = 1, \dots, n$, each with time points $0 < T_{i1} < \dots < T_{im} = T$, ($m = m_i = N_i(T)$), and increments Y_{ij} , $j = 1, \dots, m_i$. Let us first recall the likelihood process of $N(t)$:

$$V_t = \prod_{i=1}^n \left\{ \prod_{t>0} h(t)^{dN_i(t)} \cdot \exp \left(- \int_0^T h(t) dt \right) \right\},$$

and corresponding Nelson–Aalen estimate of cumulated rate $H(t)$:

$$\hat{H}(t) = \sum_{i=1}^n \int_0^t \frac{dN_i(s)}{n} = \frac{1}{n} \sum_{i=1}^n N_i(t^-) = \frac{1}{n} \sum_{i=1}^n \sum_j 1_{[T_{ij} < t]}, \quad (3)$$

see also references on counting processes, e. g. [1]. Similarly, the likelihood of rate $g(y)$ of increments distribution can be written as

$$V_y = \prod_{i=1}^n \prod_{j=1}^{m_i} \left\{ \prod_{y>0} g(y)^{dS_{ij}(y)} \cdot \exp \left(- \int_0^\infty g(y) J_{ij}(y) dy \right) \right\},$$

where $dS_{ij}(y) = 1$ just at Y_{ij} and $J_{ij}(y)$ are random indicators, $J_{ij}(y) = 1$ for $0 \leq y \leq Y_{ij}$, $J_{ij}(y) = 0$ otherwise. Hence, at fixed y , $\bar{J}(y) = \sum_{i=1}^n \sum_j J_{ij}(y)$ is the number of increments larger than y ($\bar{J}(y)$ is called the risk set in the field of survival analysis). Again, the estimator of Nelson–Aalen type for $G(y)$ yields

$$\hat{G}(y) = \sum_{i=1}^n \sum_j \int_0^y \frac{dS_{ij}(x)}{\sum_{k=1}^n \sum_l J_{kl}(x)} = \sum_{i=1}^n \sum_j \frac{1[Y_{ij} < y]}{\sum_{k=1}^n \sum_l J_{kl}(Y_{ij})}.$$

These estimators are consistent (uniformly on each bounded interval $[0, T] \times [0, Y]$) and asymptotically normal (in the sense of convergence of properly normalized residual process to the Wiener process). Estimates of hazard rates $h(t)$ or $g(y)$ are as a rule computed with the aid of kernel smoothing of increments $\Delta \hat{H}(T_{ij})$, $\Delta \hat{G}(Y_{ij})$.

4 Model with Time-Dependent Increments

Let us now consider the case that increments depend on the time of their occurrence. Hence, the process is described by hazard rates $h(t)$ and $g(y; t)$, time t acts as a covariate for hazard rate of increments. We again assume that n processes $N_i(t)$, $C_i(t)$ are observed in $[0, T] \times [0, \infty]$ fully, without censoring, so that corresponding indicators are $I_i(t) = 1$ on $[0, T]$, $J_i(y; t) = 1$ for $y \in [0, Y_i(t)]$ provided $Y_i(t)$ is an increment of i -th process at t , zero otherwise. We also assume that functions h and g are bounded, it follows that the risk set $\bar{J}(y) = \int_0^T \sum_{i=1}^n J_i(y; t) dN_i(t)$ is $O_P(n)$ uniformly for each $y \in [0, Y]$, each finite Y . In other words $\bar{J}(y)/n$ has a P -limit uniformly greater than zero and bounded, on $[0, Y]$, when n increases to infinity.

As regards the statistical analysis, the estimate of cumulated hazard rate $H(t) = \int_0^t h(s) ds$ can be obtained from the Nelson–Aalen estimator as in (3). Hazard rate of increments is a function of two variables. There exist quite general methods of its estimation, for instance the method of nonparametric estimation of doubly-cumulative hazard rate, [5] (the estimate is actually of Nelson–Aalen type w.r. to y and of kernel type w.r. to t). In the sequel, we shall consider the Cox model specification of $g(y; t)$.

4.1 Proportional Hazard Model for Increments

Let us assume that the hazard rate of distribution of increment at time t can be written as

$$g(y; t) = g_0(y) \cdot e^{b(t)}, \quad (4)$$

where $g_0(y)$ is a baseline hazard rate and $b(t)$ is a (nonparametric, in general) response function. It is seen that functions in (4) are not given uniquely, some normalization is necessary. For instance we can keep $b(t_0) = 0$ at a chosen point t_0 . Standard case deals with a parametrized function $b(t)$, nonparametric maximal likelihood can be solved via the local scoring method. The simplest approach uses a histogram-like estimator of $b(t)$, i. e. taking $\hat{b}(t) = \hat{b}_r$ constant in selected intervals \mathcal{T}_r , $r = 1, \dots, m$, dividing $[0, T]$ (while it is assumed that the actual unknown $b(t)$ is a continuous function). The constants \hat{b}_r are obtained from the maximization of the logarithm of Cox partial likelihood

$$L_p = \sum_{r=1}^m \int_0^\infty \ln \left(\frac{e^{\hat{b}_r}}{\sum_{s=1}^m e^{\hat{b}_s} \bar{J}(y, s)} \right) d\bar{S}(y, r), \quad (5)$$

where $\bar{S}(y, r)$ is now the counting process of increments in the time interval \mathcal{T}_r . Finally, cumulated baseline hazard rate $G_0(y) = \int_0^y g_0(x) dx$ is estimated with the aid of the Breslow-Crowley estimator as

$$\hat{G}_0(y) = \int_0^y \sum_{r=1}^m \frac{d\bar{S}(x, r)}{\sum_{s=1}^m e^{\hat{b}_r} \bar{J}(x, s)}. \quad (6)$$

Let us remind here that the piecewise constant function $\hat{b}(t)$ has here also the character of heterogeneity variable describing the departures of distribution of increments in certain time intervals from the baseline distribution given by $g_0(y)$.

Naturally, the assumption on proportional hazard should be verified, the tests of proportionality of two subsamples as well as the goodness-of-fit test for Cox model are available, [1].

As regards the asymptotic properties, i.e. the situation when $n \rightarrow \infty$, it is as a rule assumed that the number of histogram intervals $m = m_n \rightarrow \infty$ and $n/m_n \rightarrow \infty$, too. Under certain more-less technical conditions the consistency of estimation can be proven. The theory for parametrized case is already well developed (see again [1]). The asymptotics for the more complex nonparametric case can use several sources. One of them is based on already mentioned results of [5] and their specific approach to solution for a general model. Another way leads from the theory of consistent spline models derived by C. Stone, e.g. [6]. The histogram approximation is actually a special case of spline model.

5 Example

As an example, we analyzed the processes of credit cards payments at a hotel. One process corresponded to payments on one day, data are from $n = 90$ days, $t \in [0, 24]$ hours. Figure 1 shows a selection of realized processes $N_i(t)$ and $C_i(t)$, together with estimated cumulated rates $\hat{H}(t)$ and $\hat{K}(t)$, actually the

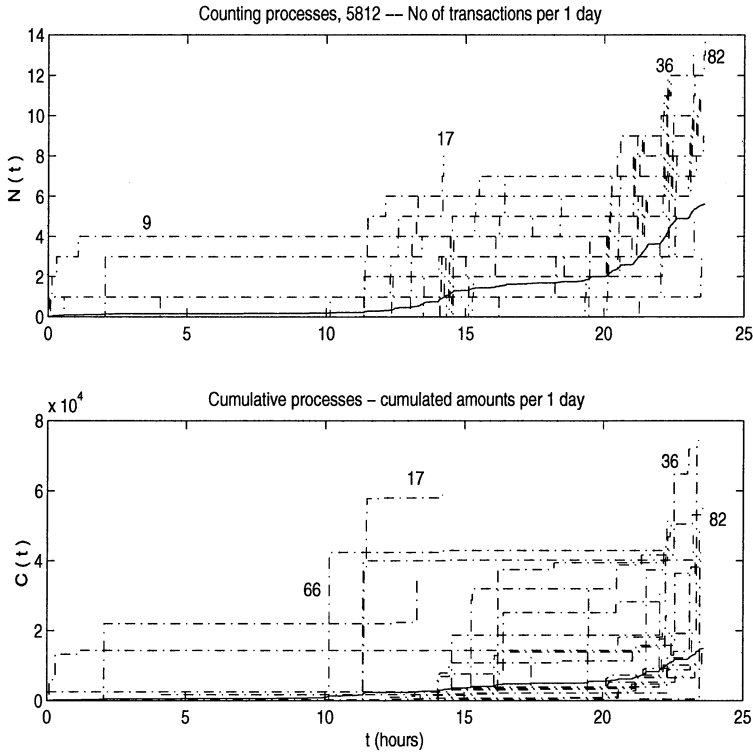


Fig. 1. Observed processes

means from observed trajectories. Certain interesting trajectories are denoted by their numbers. The estimates indicate that the frequency of increments as well as their magnitude depend on time. No other covariates are considered, we assume that the model of preceding section describes the case sufficiently. Graphical test confirmed a good fit of the model to our data.

The time period $[0, 24]$ hours was divided to 24 one-hour intervals. The piecewise constant estimator \hat{b}_r has been obtained from (5), then it was smoothed by a moving window. The result is on the lower subplot of Figure 2. We kept $b_1 = 0$ in order to guarantee uniqueness of solution. Further, the cumulated baseline rate estimate $\hat{G}_0(y)$ has been computed along (6), its plot is in the upper subplot. Finally, we constructed approximate upper prediction lines both for numbers and cumulation of increments. For all trajectories their actual times and increments are available and the model has been just estimated. Hence, actual behaviour of observed trajectories can be at once compared with their expected behaviour derived from the model. More precisely, if a process has times T_j and increments $Y_j = Y(T_j)$, and model with $H(t)$ and $G(y; t)$ is the right one, then times $\tau_j = H(T_j)$ should be the times of Poisson(1) process and values $\gamma_j = G(Y_j, T_j)$ should be the values corre-

sponding to $\text{Exp}(1)$ distribution. Therefore, the scale of each trajectory $C_i(t)$ can be transformed to the scale of compound Poisson $(1,1)$ process. The results of such transformation, for four interesting trajectories, is displayed in Figure 3, in the first subplot. The dashed line is the 95 % prediction line $u + vt$, $u = 9.308$, $v = 1.4$ (computed in section 2).

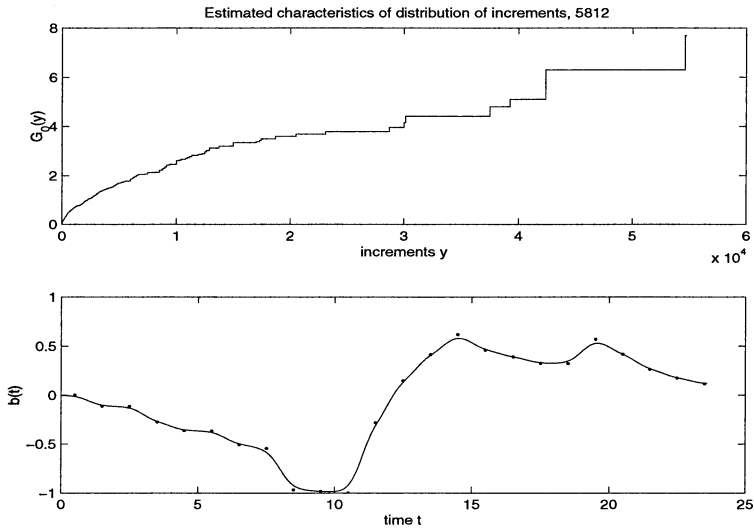


Fig. 2. Estimated cumulated baseline rate $G_0(y)$ and response function $b(t)$

Simultaneously, in subplot 2 of Figure 3, we compared also corresponding counting processes with the upper 95% prediction line $u + vt$, $u = 8.17$, $v = 1.2$ (again one of those derived in section 2). The diagnostics revealed several trajectories with (not extremely) outlied character consisting mostly in that the cumulated sum crossed the prediction line. Naturally, the more convenient variant of diagnostics should use the cross-validation, i.e. the model evaluated only from trajectories not selected for the test.

6 Conclusion

The main purpose of the paper was to offer a simple model for the cumulative processes consisting in the combination of the counting process with random increments dependent on it and to show an application to the analysis of stream of financial transactions. Successful use of such models requires the development of the methods for estimation of the model characteristics and also the methods for the prediction of process behaviour under different conditions. Then we are also able to classify the processes and, eventually, to detect atypical ones. The practical application for instance to the fraud detection problem is quite straightforward.

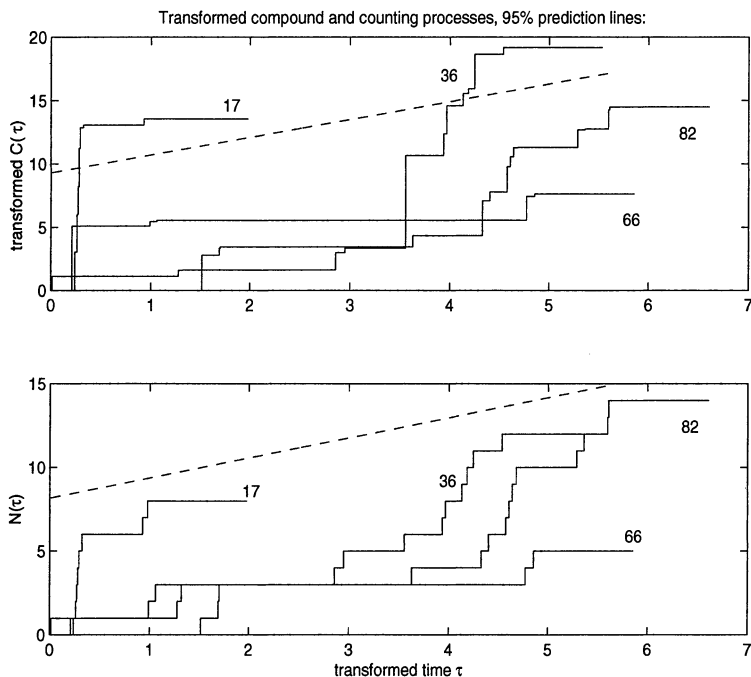


Fig. 3. Selected trajectories, comparison with prediction lines

Acknowledgement: The research has been supported by the project of GA ĆR No 402/01/0539.

References

1. Andersen, P. K., Borgan, O., Gill, R. D., Keiding, N. (1993). Statistical Models Based on Counting Processes. Springer, New York.
2. Asmussen, S. (2000). Ruin Probabilities. World Scientific, Singapore.
3. Embrechts, P., Klüppelberg, K., and Mikosch, T. (1997). Modeling Extremal Events. Springer, Berlin.
4. Grandell, J. (1997). Mixed Poisson Processes. Chapman and Hall, London.
5. McKeague, I. W., Utikal, K. J. (1990). Inference for a nonlinear counting regression model. *Annals Statist.* 18, 1172–87.
6. Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *Annals of Statist.* 22, 118–184.
7. Volf, P. (2000). On cumulative process model and its statistical analysis. *Kybernetika* 36, 165–176.

Structural Optimization in Aircraft Engineering using Support Vector Machines

Peter Kaletta¹, Klaus Wolf¹, and Andreas Fischer²

¹ Technische Universität Dresden, Institute of Aerospace Engineering,
01062 Dresden, Germany, Email: {kaletta,wolf}@lft.mw.tu-dresden.de

² Technische Universität Dresden, Institute of Numerical Mathematics,
01062 Dresden, Germany, Email: fischer@math.tu-dresden.de

Abstract. The minimum weight design of structures made of fiber reinforced composite materials leads to a class of discrete optimization problems for which evolutionary algorithms (EAs) are well suited. Based on these algorithms the optimization tool package GEOPS has been developed at TU Dresden.

For each structure generated by an EA the structural response has to be evaluated. This is often based on a finite element analysis, which results in high computational efforts for each single structure. Typical runs of EAs require the evaluation of thousands of structures. Thus, an efficient approximation of the structural response could improve the performance considerably. To achieve this aim, the use of a support vector machine (SVM) is suggested in this paper.

As an example for a typical aircraft structure, a stiffened composite panel under compressive and shear loading is considered. Buckling as well as strength constraints are taken into account. The SVM is trained on geometrical and material data. Representing the design space of composite panels by ABD matrices turned out to be a valuable means for obtaining well trained SVMs.

1 Introduction

Primary objectives concerning the development of next generation commercial transport aircrafts are to reduce the manufacturing costs and the structural weight considerably. A promising way to achieve these aims is to extend the application of fiber-reinforced composite materials. The use of laminated composites leads to a substantial increase in design parameters. Hence, minimum weight solutions for more complex structural components such as stiffened fuselage panels and wing skins can be obtained only by using numerical optimization methods combined with finite element codes.

The optimization of stiffened panels made of laminated composites is characterized by a combination of continuous and discrete design variables. Typical examples of discrete parameters are the ply thickness and the ply orientations. Caused by production constraints the plies of the skin and stiffener laminates in real structures usually have to be chosen from a finite number of thicknesses and a limited set of ply orientations such as 0, 90 and ± 45 degrees. The number and type of stiffeners are discrete parameters, while the stiffener height is a continuous design variable. These mostly discrete design

variables restrict the use of traditional continuous optimization techniques. In the last decades techniques based on natural evolution principles have been introduced to deal with this type of optimization problems. A combination of such techniques [10], namely of evolution strategies (ES) and genetic algorithms (GA), is implemented in the optimization package *Genetic and Evolutionary Optimization of Structures* (GEOPS) developed at the Institute of Aerospace Engineering [3].

The optimization of the laminate stacking sequence using evolutionary algorithms is the subject of several recent studies [2,5,8,9,11]. In [11] sandwich structures with composite facesheets are investigated whereas panels with discrete stiffeners are dealt with in the other references. Except the papers related to GEOPS [2,3,11] only GA-type algorithms have been used.

The optimization of stiffened composite panels involves constraints such as strength failure or buckling. Frequently, closed form solutions (as given in [4]) are used to compute constraint values. However, this approach is restricted to simple panels. To obtain a reliable evaluation of more complex structures a finite element (FE) code has to be employed. In [2] such a numerical analysis is suggested for all structures occurring during the whole evolutionary optimization process. Of course, this can be very expensive in terms of computation time.

To overcome this drawback several concepts have been developed recently. In [9] a FE analysis is carried out for a number of randomly generated structures. Then, a neural network provides approximations for constraint values. A composite wing structure is dealt with in [5] by a two-level design procedure. Before entering the upper level an approximation of the buckling load by means of a response surface method is provided in the lower level. Then, for some structures generated in the upper level a FE analysis is performed, but without feedbacks to the lower level.

In this paper a new approximation concept for saving FE computations will be suggested. A support vector machine (SVM) is used to separate the design space into a feasible part (where all constraints are satisfied) and an infeasible one. Based on a certain distance to the separating surface provided by the SVM an approach to define the fitness function used in the evolutionary algorithm is presented. Although SVMs have been widely investigated in recent years (for an overview see [1,7]) we are not aware of applications for structural optimization.

2 Structural Optimization and Evolutionary Algorithms

The aim to improve or to optimize a structure presumes the potential for changes of the structure. Here, this potential is expressed by means of *design variables* x_i with $i = 1, \dots, n$. The *design vector* $\mathbf{x} := (x_1, \dots, x_n)^T$ controls the geometry and material properties of the structure. Depending on its role

the design variable x_i can take either continuous or discrete values in a certain range, i.e., $x_i \in D_i \subseteq \mathbb{R}$ for $i = 1, \dots, n$. Then,

$$\mathcal{D} := D_1 \times \dots \times D_n \subset \mathbb{R}^n$$

denotes the *design space*. Any element of \mathcal{D} corresponds to a particular structure and vice versa. Of course, there are other restrictions a structure has to satisfy. They stem from structural limits like allowable stress and strain, maximum deformations and buckling loads. If a structure $\mathbf{x} \in \mathcal{D}$ satisfies these restrictions it is called *feasible* and belongs to the set $\mathcal{F} \subset \mathcal{D}$ of feasible structures. To compare structures to each other an *objective function* $f : \mathcal{D} \rightarrow \mathbb{R}$ is used, for example the weight or production cost.

The *structural optimization problem* of finding a feasible structure \mathbf{x}^* which minimizes (or maximizes) the objective function f over the feasible set \mathcal{F} is, for short, written as

$$f(\mathbf{x}) \rightarrow \min \quad \text{subject to} \quad \mathbf{x} \in \mathcal{F}. \quad (1)$$

To apply an evolutionary algorithm (that generates both feasible and infeasible structures in the design space \mathcal{D}) problem (1) is often replaced by a problem over \mathcal{D} . Therefore, the violation of the restriction $\mathbf{x} \in \mathcal{F}$ is penalized by means of a certain penalty function $p : \mathcal{D} \rightarrow [0, \infty)$ and problem (1) is replaced by

$$\phi(\mathbf{x}) := f(\mathbf{x}) + p(\mathbf{x}) \rightarrow \min \quad \text{subject to} \quad \mathbf{x} \in \mathcal{D}. \quad (2)$$

An optimization aims at finding a design vector \mathbf{x} (i.e., a structure) with a minimal value of the *fitness function* $\phi : \mathcal{D} \rightarrow \mathbb{R}$.

Evolutionary algorithms are promising methods to treat complex problems with discrete and continuous design variables. They mimic the biological evolution process. In each iteration a population of individuals is generated. Here, an individual represents a particular structure \mathbf{x} in the design space \mathcal{D} . As the iteration process goes on the populations evolve, taking into account that certain characteristics of the parent population are passed to the offspring population. Structures with better fitness function values have better chances to survive or to pass certain characteristics to an offspring population.

Our investigation is based on the implementation of an evolutionary algorithm in the software package GEOPS [3]. The structure of this algorithm is given in Fig. 1. A part of the offspring population is created by using ES operators and another part by means of GA operators. The main differences between these two types of operators lie in the handling of design variables. ES operators directly work on the design variables whereas GA operators are applied to a coded form of the design variables as a binary string. For more details about the optimization with ES and GA see [2,3,10,11].

The algorithm shown in Fig. 1 generates many populations, resulting in a huge number of structures. For each structure the violation of the constraints

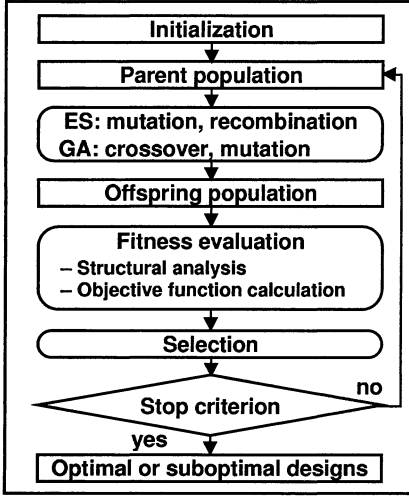


Fig. 1. Evolutionary optimization

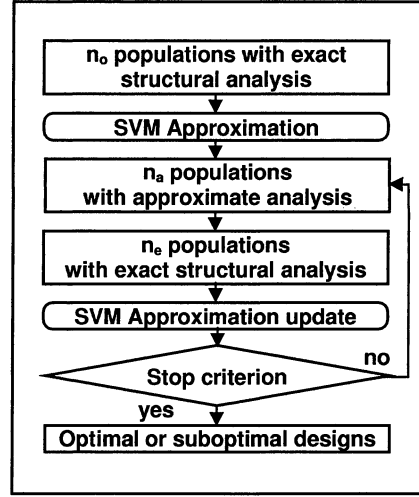


Fig. 2. Approximation concept

has to be evaluated, often by means of an expensive FE analysis. To reduce the resulting computational effort significantly we suggest an approximation of the overall constraint violation of every individual that occurs within certain intermediate populations. The approximation is done by means of a support vector machine (SVM), which is described in Section 3. At the beginning of the optimization process n_0 populations with exactly computed constraint violations are generated. Then, n_a populations with approximate evaluation of the overall constraint violation and n_e populations with exactly computed constraint violations alternate, see Fig. 2. Of course, after each cycle with exact evaluation the training data for the SVM is updated to improve the approximation quality.

3 SVM Classification and Approximation Concept

Any function $c : \mathcal{D} \rightarrow \{-1, 1\}$ divides the design space \mathcal{D} into the disjoint sets

$$\mathcal{D}_+ := \{\mathbf{x} \in \mathcal{D} \mid c(\mathbf{x}) = 1\} \quad \text{and} \quad \mathcal{D}_- := \{\mathbf{x} \in \mathcal{D} \mid c(\mathbf{x}) = -1\}.$$

The function c is chosen so that $\mathcal{D}_+ = \mathcal{F}$ is the set of feasible and $\mathcal{D}_- = \mathcal{D} \setminus \mathcal{F}$ the set of infeasible design vectors. Only for m training data points $\mathbf{p}_i \in \mathcal{D}$ ($i = 1(1)m$) the values $c(\mathbf{p}_i)$ are assumed to be known. Then, the set of training data points, denoted by T , decomposes into the disjoint subsets

$$T_+ := \{\mathbf{p}_i \mid c(\mathbf{p}_i) = 1, i = 1(1)m\} \quad \text{and} \quad T_- := \{\mathbf{p}_i \mid c(\mathbf{p}_i) = -1, i = 1(1)m\}.$$

Based on the training data information a SVM classifies any point $\mathbf{x} \in \mathcal{D}$, i.e., provides an estimate for $c(\mathbf{x})$. In some cases, this value might be false. In

principle, a SVM computes a hyperplane in \mathbb{R}^n to separate the sets T_+ and T_- . In general, such a hyperplane does not need to exist. Therefore, \mathbb{R}^n is mapped into a higher dimensional space H so that the images of T_+ and T_- in H can be separated by a hyperplane in H . Among all such hyperplanes a SVM chooses one with some maximal distance to the images of both T_+ and of T_- . This hyperplane in H then induces a surface in \mathbb{R}^n that separates T_- and T_+ and divides \mathcal{D} into two subsets. With the (m, n) -matrix \mathbf{P} containing \mathbf{p}_i^T as rows and the diagonal (m, m) -matrix \mathbf{C} with the diagonal entries $c(\mathbf{p}_i)$ ($i = 1(1)m$), any pair $(\mathbf{u}, \gamma) \in \mathbb{R}^{n+1}$ defines a surface

$$\{\mathbf{x} \in \mathbb{R}^n \mid K(\mathbf{x}, \mathbf{P}^T)\mathbf{C}\mathbf{u} - \gamma = 0\}. \quad (3)$$

Here, $K(\mathbf{X}, \mathbf{Z}) := (k(\mathbf{x}_i, \mathbf{z}_i))$ depends on $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_\ell)^T \in \mathbb{R}^{\ell \times n}$ and $\mathbf{Z} := (\mathbf{z}_1, \dots, \mathbf{z}_l) \in \mathbb{R}^{n \times l}$. For $k : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ the Gaussian kernel

$$k(\mathbf{x}, \mathbf{z}) := \exp(-\mu \|\mathbf{x} - \mathbf{z}\|_2^2) \quad \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^n \quad (4)$$

is chosen with a fixed parameter $\mu > 0$. The pair (\mathbf{u}^*, γ^*) is obtained as solution of the following SVM (with $\mathbf{e} := (1, \dots, 1)^T \in \mathbb{R}^m$)

$$\begin{aligned} \nu_1 \|\mathbf{y}_1\|_2^2 + \nu_2 \|\mathbf{y}_2\|_2^2 + \gamma^2 + \|\mathbf{u}\|_2^2 &\rightarrow \min \\ \text{s.t. } \mathbf{C}(K(\mathbf{P}, \mathbf{P}^T)\mathbf{C}\mathbf{u} - \gamma\mathbf{e}) + \mathbf{y} &\geq \mathbf{e}. \end{aligned} \quad (5)$$

The vector $\mathbf{y} \in \mathbb{R}^m$ of slack variables allows some training data points to be misclassified in order to enable a better separation of the remaining ones. The subvector \mathbf{y}_1 (\mathbf{y}_2) contains the components y_i of \mathbf{y} for which \mathbf{p}_i is feasible (infeasible). Smaller misclassification errors $\|\mathbf{y}_1^*\|_2^2$ and $\|\mathbf{y}_2^*\|_2^2$ can be achieved by larger penalization parameters $\nu_1, \nu_2 > 0$ in (5).

Having a solution $(\mathbf{u}^*, \gamma^*, \mathbf{y}^*)$ of the SVM (5) the surface (3) yields an approximate classification of design vectors $\mathbf{x} \in \mathcal{D}$ into feasible and infeasible ones. If $K(\mathbf{x}, \mathbf{P}^T)\mathbf{C}\mathbf{u}^* - \gamma^* \geq 0$, then \mathbf{x} is regarded as feasible, otherwise as infeasible. Moreover, we would like to approximate the overall constraint violation for any design vector $\mathbf{x} \in \mathcal{D}$. For this purpose, the expression

$$d(\mathbf{x}) := \max\{0, -(K(\mathbf{x}, \mathbf{P}^T)\mathbf{C}\mathbf{u}^* - \gamma^*)\} \quad \forall \mathbf{x} \in \mathcal{D}.$$

is involved. Then, $d(\mathbf{x}) = 0$ if \mathbf{x} is classified as feasible. Otherwise, $d(\mathbf{x}) > 0$ and the value $d(\mathbf{x})$ can be regarded as a measure for the distance of \mathbf{x} to the set \mathcal{F} of feasible design vectors. Therefore, $d(\mathbf{x})$ estimates the overall constraint violation and will be used to define the penalty function p by

$$p(\mathbf{x}) := \exp(\alpha d(\mathbf{x})) - 1 \quad \forall \mathbf{x} \in \mathcal{D}$$

with a fixed $\alpha > 0$. Thus, the fitness function ϕ (see (2)) is well defined for all design vectors \mathbf{x} generated in a phase of approximation. In a phase where the constraint violations are computed exactly these violations are explicitly used to define p .

4 The Design Vector

Every structure in the design space \mathcal{D} is represented by a design vector \mathbf{x} . The components of the vector provide information both on geometrical data (like the number of stiffeners or their height) and on material data of the laminates used for the skin and the stiffeners. A laminate is a sequence of single plies arranged on top of each other. Commonly, these plies are reinforced by unidirectionally orientated fibers embedded in a polymer.

For a successful application of the SVM the discrete structure of the laminates is replaced by an integral model. Based on the classical lamination theory [12] the material data are represented by the so-called ABD-matrix. This stiffness-matrix relates the forces N_i and the moments M_i applied to a laminate to the mid-plane strains ε_{i0} , γ_{xy0} and the curvatures κ_i (see Fig. 3):

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} \quad (6)$$

The A and D submatrices are the extensional and flexural stiffness matrices, respectively. Submatrix B is called the bending-extension coupling matrix. Making use of the symmetry of the ABD matrix each laminate is represented by 18 parameters.

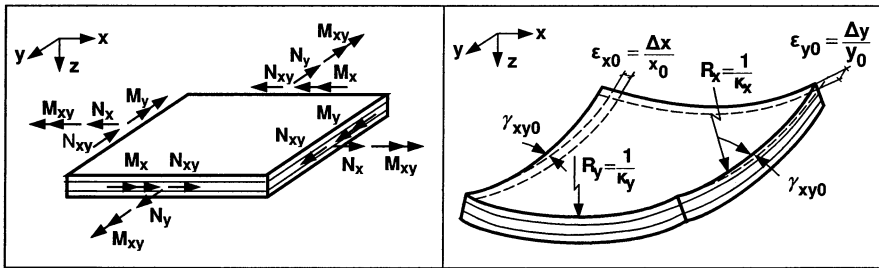


Fig. 3. Non-deformed and deformed laminate element

5 Numerical Test Example

A blade stringer stiffened composite panel (length 1500mm, width 1047mm) under compressive and shear loading ($N_x = -132\text{N/mm}$, $N_{xy} = 220\text{N/mm}$) is considered. Skin and stiffeners consist of unidirectional graphite/epoxy prepregs with $0, \pm 45$ and 90 degree orientations. The design vectors contain number and height of stiffeners and the material data for the laminates of skin and stiffeners. To define the feasible set $\mathcal{F} \subset \mathcal{D}$ strength and buckling constraints are taken into account.

The influence of the SVM parameter ν_1 (see (5)) and of the Gaussian kernel parameter μ (see (4)) on the rate of misclassification of design vectors by the SVM has been studied (ν_2 was always set to $0.05 \nu_1$). After $n_0 := 12$ exactly evaluated populations two cycles each with $n_a := 10$ approximate and $n_e := 5$ exact populations follow. At the end, 3 exact test populations are generated to check the quality of the approximation provided by the last SVM. In every population, 24 offspring individuals are created based on the design information of 10 selected parent individuals. GEOPS also features techniques to avoid the loss of good feasible structures during approximate populations, to detect certain wrongly classified structures, and to amplify the penalization.

The misclassification rate over all approximate populations (Table 1) and over the last 3 test populations (Table 2) is given as a mean percentage over 10 runs of GEOPS. Based on this study (see [6]), the parameters $\nu_1 = 10^4$

Table 1. Misclassification rate in the approximate populations in %

ν_1	$\mu = 0.002$	0.2	5	50
10		12.4	21.8	38.8
100		7.6	19.1	40.9
10^4	14.0	8.1	21.8	40.3
10^6	9.4	8.6	19.6	38.2

Table 2. Misclassification rate in the test populations in %

ν_1	$\mu = 0.002$	0.2	5	50
10		6.2	9.0	14.9
100		6.5	9.6	17.6
10^4	7.9	3.2	7.2	18.1
10^6	6.0	6.5	8.2	18.5

and $\mu = 0.2$ are chosen for further investigations. The best fitness function values of each population obtained in 5 runs of GEOPS are shown in Fig. 4 for exactly evaluated populations and in Fig. 5 for cycles with exact and approximate populations. Regardless whether approximate populations are

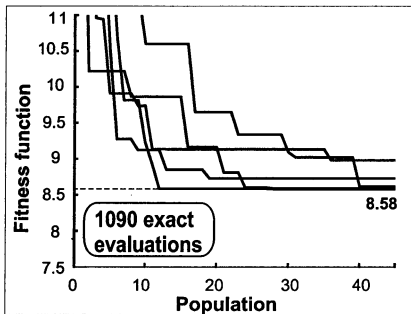


Fig. 4. GEOPS with exact evaluations only

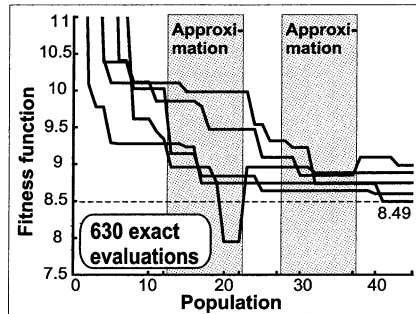


Fig. 5. GEOPS with approximate evaluations by means of SVMs

involved or not the optimization processes lead to structures with similar weights. Of course, there is no guarantee of optimality, i.e., a feasible structure with less weight might exist. In the runs with SVM based approximate populations the number of exactly evaluated structures reduces from 1090

to 630. Further substantial savings of computation time are expected from developing more sophisticated combinations of exactly and approximately evaluated structures.

References

1. Christianini N., Shawe-Taylor J (2000) An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge
2. Kaletta P., Wolf K. (2000) Optimization of composite aircraft panels using evolutionary computation methods. In: Proc. 22nd ICAS Conference, ICAS, Les Mureaux Cedex, 411.1–411.10
3. Kaletta P., Wolf K. (2000) Strukturoptimierung von Flugzeugstrukturen mit Hilfe von evolutionären Algorithmen. In: Jahrbuch 2000 der Deutschen Gesellschaft für Luft- und Raumfahrt, DGLR, Bonn, Vol. 3, 1–10
4. Kogiso N., Watson L.T., Gürdal Z., Haftka R.T., Nagendra S. (1994) Design of composite laminates by a genetic algorithm with memory. *Mechanics of Composite Materials and Structures* 1:95–117
5. Liu B., Haftka R.T. Akgün (2000) Two-level composite wing structural optimization using response surfaces. *Structural and Multidisciplinary Optimization* 20:87–96
6. Löschner K. (2003) SVM Based Approximation in Structural Optimization. Diploma Thesis, Dresden University of Technology, Dresden, in preparation
7. Mangasarian O.L. (2003) Data mining via Support Vector Machines. In: Sachs E.W., Tichatschke R. (Eds.) *System Modeling and Optimization XX*, Kluwer Academic Publishers, Boston, 91–112.
8. Nagendra S., Jestin D., Gürdal, Haftka R.T. Watson L.T. (1996) Improved genetic algorithm for the design of stiffened composite panels. *Computers & Structures* 58:543–555
9. Ryoo J., Hajela P. (2001) Handling variable string lengths in GA based structural topology optimization. In: Proc. 42nd Structures, Structural Dynamics and Material Conference, AIAA, Reston, AIAA 2001-1629: 1–11
10. Schwefel H.-P. (1994) *Evolution and Optimum Seeking*. Wiley-Interscience, New York
11. Wolf K. (2001) Optimization of composite sandwich panels using evolutionary computation methods. In: Proc. 42nd Structures, Structural Dynamics and Material Conference, AIAA, Reston, AIAA 2001-1277: 1–11
12. Zenkert D. (1995) *An Introduction to Sandwich Construction*. Chameleon Press Ltd, London

Zeitformen in einer probabilistischen Konditionallogik

Elmar Reucher und Wilhelm Rödder

Lehrstuhl für Betriebswirtschaftslehre,
insb. Operations Research
FernUniversität in Hagen

elmar.reucher@fernuni-hagen.de
wilhelm.roedder@fernuni-hagen.de

Zusammenfassung (Probabilistische) Konditionale stellen ein Kommunikationsinstrument dar, mit dem Wissen in einer reichen Sprache vermittelt werden kann. Ziel dieses Artikels ist es aufzuzeigen, wie die Konditionalsprache erweitert werden kann, um auch Sprachgefüge mit temporalem Kontext auszudrücken. Zur Verarbeitung des Wissens dient das axiomatisch fundierte Entropieprinzip. Anhand eines Beispiels mittlerer Größe werden die theoretischen Überlegungen in der Expertensystem-Shell SPIRIT näher illustriert.

1 Einführung

In Erweiterung der klassischen Aussagenlogik etablierte sich in den letzten Jahren eine probabilistische Konditionallogik, die sich zur Inferenz des Prinzips Maximaler Entropie (MaxEnt) bedient [1], [6]. Zur Berücksichtigung temporal-logischer Aspekte finden sich u.a. Ansätze erweiterter Modallogiken [3], Intervalllogiken [4] oder Verzweigungslogiken [10]. In dem vorliegenden Beitrag wird der MaxEnt - Inferenzprozess um temporale Elemente angereichert, die weitgehend dem menschlichen Sprachgefühl entsprechen. Der gesamte Ansatz basiert auf der Verwendung von Zeitkonditionalen, mit deren Aktivierung oder Inaktivierung der gesamte Zeitstrahl von Zeiten in der Vergangenheit bis hin in die Zukunft durchlaufen werden kann. Die Modellierung erfolgt in der Expertensystem-Shell SPIRIT, mittels derer der Zeitprozess anhand eines Beispiels demonstriert wird, vgl. [9].

2 Grundlagen

2.1 Probabilistische Konditionale

Es sei \mathcal{L} eine propositionale Sprache, bestehend aus einer endlichen Menge endlichwertiger Variabler $\mathcal{V} = \{V_1, \dots, V_n\}$ mit Werten $V_j = v_j$. Propositionen in \mathcal{L} sind wohlgeformt aus Literalen $V_j = v_j$, aus den Junktoren \wedge (und), \vee (oder), \neg (nicht) und entsprechenden Klammern; solche propositionalen Ausdrücke werden mit A, B, C, \dots bezeichnet. Bei Konjunkten wird

der Junktoren oft unterdrückt: $AB = A \wedge B$. Vollständige oder einfache Konjunktionen von Literalen schreiben wir als ungeordnete Tupel, wie zum Beispiel $\mathbf{v} = v_1 \dots v_n$.

$|$ ist der binäre Konditionaloperator. Ausdrücke der Form $B|A$ heißen Konditionale oder Regeln, sie bilden die Sprache $\mathcal{L}|\mathcal{L}$. Propositionen $A \in \mathcal{L}$ werden mit Konditionalen $A|T$ identifiziert, wobei T irgendeine Tautologie ist. Konditionale können wiederum konditioniert werden. Der hier benötigte einfache Fall ist der, der Konditionierung unter einer Präposition $(B|A)|C$ mit der Bedeutung $(B|A)|C = B|AC$. Über weitere Eigenschaften von $\mathcal{L}|\mathcal{L}$ vgl. [1]. Den Konditionalen werden Wahrscheinlichkeiten $x \in [0; 1]$ zugewiesen. Somit besteht die Syntax aus Objekten der Form $B|A[x]$, $A, B \in \mathcal{L}$ und $x \in [0; 1]$. Ein semantisches Modell ist eine Verteilung P auf \mathcal{L} . In P ist ein Sachverhalt $B|A[x]$ gültig, $P \models B|A[x]$, genau dann wenn $P(BA) = x \cdot P(A)$. Sind in P mehrere Konditionale $\mathcal{R} = \{B_i|A_i[x_i], i = 1, \dots, I\}$ gültig, schreiben wir kurz $P \models \mathcal{R}$. Sind sämtliche Konditionale $B_i|A_i[x_i]$ unter C zu konditionieren, also $B_i|A_iC[x_i]$, notiert man das einfach als $\mathcal{R}|C$.

2.2 Das Entropieprinzip

Wissen in Form einer Menge von Sachverhalten $\mathcal{R} = \{B_i|A_i[x_i], i = 1, \dots, I\}$ wird in diesem Beitrag entropieoptimal durch Lösen der Aufgabe

$$P^* = \arg \max_{\mathbf{v}} H(Q) = - \sum_{\mathbf{v}} Q(\mathbf{v}) \cdot \log_2 Q(\mathbf{v}) \quad (1)$$

u. d. N.: Q erfüllt \mathcal{R} ($Q \models \mathcal{R}$)

verarbeitet [6]. Das Prinzip *Maximaler Entropie* haben verschiedene Autoren unabhängig voneinander axiomatisch fundiert; man vergleiche hierzu [2], [7]. Die auf diese Weise erzeugte Verteilung P^* ist gerade jene unter allen Verteilungen Q , in der alle in \mathcal{R} explizit formulierten - und keine weiteren - Abhängigkeiten repräsentiert sind. Die in (1) formulierte Optimierungsaufgabe garantiert somit eine informationstreue Wissensverarbeitung. Liegen nun situativ gültige Sachverhalte \mathcal{R}_E vor, so wird das bekannte Wissen aus P^* durch Lösen der Aufgabe

$$P^{**} = \arg \min_{\mathbf{v}} R(Q, P^*) = \sum_{\mathbf{v}} Q(\mathbf{v}) \cdot \log_2 \left(\frac{Q(\mathbf{v})}{P^*(\mathbf{v})} \right) \quad (2)$$

u. d. N.: $Q \models \mathcal{R}_E$

informationstreu an das neue Wissen aus \mathcal{R}_E adaptiert. Man vergleiche hierzu auch [6]. Im folgenden Abschnitt wird gezeigt, wie die probabilistische Konditionallogik um temporale Elemente erweitert werden kann.

3 Abbildung von Zeitformen durch Konditionale

Wir betrachten eine Wissensdomäne über einem Zeithorizont, wie in Abbildung 1 dargestellt. Dabei bezeichnet t den Zeitpunkt der Gegenwart, $t - 2$

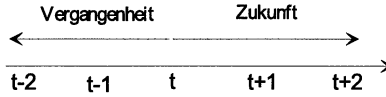


Abbildung1. Modellhorizont

und $t - 1$ bezeichnen Zeitpunkte der Vergangenheit sowie $t + 1$ und $t + 2$ Zeitpunkte in der Zukunft. Eine Verallgemeinerung auf mehr als zwei vergangene oder zukünftige Zeitpunkte ist trivial, soll aber hier aus Gründen der Übersichtlichkeit unterbleiben. Es sei $\mathcal{R}_{t+\tau}$ eine Menge von Sachverhalten, die zum Zeitpunkt $t + \tau$ mit $\tau = -2, -1, 0, 1, 2$ bekannt wurden, werden oder (vermutlich) in Zukunft zufließen werden. Desweiteren bezeichnen $W_{t+\tau}$ binäre (Zeit-) Variable. Dann sind $\mathcal{R}_{t+\tau}|W_{t+\tau}$ unter den $W_{t+\tau}$ zeitkonditionierte Sachverhalte, die im folgenden kurz *Zeitkonditionale* heißen.

Ziel ist es nun, eine Wissensbasis zu erzeugen, die eine Erinnerung an vergangene Wissenszustände und eine Vorwegnahme zukünftiger Zustände – unter Einbeziehung vermutlich richtiger noch zufließender Sachverhalte – gestattet. Dabei besteht das Wissen zum Zeitpunkt $t + \tau$ aus der Aggregation aller bis dahin erlernten Sachverhalte $\mathcal{R}_{t-2} \cup \dots \cup \mathcal{R}_{t+\tau}$. Dazu setzen wir voraus, dass sämtliche bis in $t + \tau$ erworbenen Sachverhalte untereinander konsistent, d.h. widerspruchsfrei sind. Für den Fall widersprüchlicher Sachverhalte geben die Autoren in [5] eine Methode an, wie Inkonsistenzen aufgelöst werden können. Hierauf wird in dem vorliegenden Artikel aber nicht näher eingegangen.

Der temporale, entropieoptimale Wissensverarbeitungsprozess vollzieht sich nun in drei Schritten.

1. Formulierung von Zeitkonditionalen

$$\begin{aligned} t - 2 &: (\mathcal{R}_{t-2}|W_{t-2}) \\ t - 1 &: (\mathcal{R}_{t-2}|W_{t-2} \cup \mathcal{R}_{t-1})|W_{t-1} \\ t &: ((\mathcal{R}_{t-2}|W_{t-2} \cup \mathcal{R}_{t-1})|W_{t-1} \cup \mathcal{R}_t)|W_t \\ &\vdots \end{aligned}$$

2. Abbildung des chronologischen Ablaufs

$$\mathcal{R}_c := \{W_{t+\tau-1}|W_{t+\tau} [1.0] \} \text{ für } \tau = -1, \dots, 2.$$

In Verbindung mit den Zeitkonditionalen unter 1. wird mit den Regeln \mathcal{R}_c sichergestellt, dass sowohl die zum Zeitpunkt $t + \tau$ erworbenen Sachverhalte als auch sämtliche zeitlich früher erworbenen berücksichtigt werden. Die Wissensverarbeitung erfolgt im nächsten Schritt.

3. Lösen der (temporalen) Inferenzaufgabe(n)

$$P_{t\pm 2}^* = \arg \max H(Q) \quad (3)$$

$$\text{u. d. N. : } Q \models \mathcal{R}_{t-2} | W_{t-2} \cup (\mathcal{R}_{t-2} | W_{t-2} \cup \mathcal{R}_{t-1}) | W_{t-1} \cup \dots \cup \mathcal{R}_c$$

Die Verteilung $P_{t\pm 2}^*$ birgt das über die einzelnen Zeitstufen $t - 2$ bis $t + 2$ erworbene Gesamtwissen, aus der sich nun jeder Wissensstand zum Zeitpunkt $t + \tau$ mit $\tau = -2, \dots, 2$ abrufen lässt, indem die Zeitvariable $W_{t+\tau}$ aktiviert und $W_{t+\tau+1}$ ausgeblendet wird. Formal entspricht das der Lösung der Aufgabe

$$P_{t+\tau}^* = \arg \min R(Q; P_{t\pm 2}^*) \quad (4)$$

$$\text{u. d. N. : } Q \models W_{t+\tau} [1.0] \cup W_{t+\tau+1} [0.0].$$

(4) ist ein Spezialfall von (2), allerdings erfolgt die Lösung in der Expertensystem-Shell SPIRIT durch einfaches Anklicken der Werte $W_{t+\tau} = 1$ und $W_{t+\tau+1} = 0$, vgl. auch Abbildungen 2 bis 5.

Um beispielsweise den gegenwärtigen Wissensstand P_t^* ($\tau = 0$) abzurufen, wird die entsprechende Zeitvariable W_t aktiviert ($W_t [1.0]$), was wegen des im 2. Schritt formulierten chronologischen Ablaufs auch W_{t-2} und W_{t-1} aktiviert. Zudem wird die Zeitvariable W_{t+1} ausgeblendet ($W_{t+1} [0.0]$), wodurch sich die Zeitvariable W_{t+2} mit ausblendet. Durch das Aktivieren und Ausblenden der Zeitvariablen wird sichergestellt, dass in P_t^* nur die zum Zeitpunkt t erworbenen Sachverhalte \mathcal{R}_t und die zeitlich früher erworbenen $\mathcal{R}_{t-2} \cup \mathcal{R}_{t-1}$ gelten; alle zu späteren Zeitpunkten noch erfahrbaren Sachverhalte \mathcal{R}_{t+1} und \mathcal{R}_{t+2} bleiben unberücksichtigt. Entsprechend lassen sich Wissensstände auch zu allen anderen Zeitpunkten innerhalb des Modellhorizonts abrufen.

Die etwas aufwendige Indizierung $t + \tau$ wurde gewählt um zu verdeutlichen, dass auch t eine sich im Zeitablauf verändernde Variable ist. Mit dem Übergang $t = t + 1$ wird natürlich τ zu $\tau - 1$.

Im nächsten Abschnitt werden die theoretischen Überlegungen anhand eines Beispiels erläutert. Statt der aufwendigen Relativindizierung werden jetzt einfach die Zeitpunkte $-2, -1, 0, 1, 2$ betrachtet.

4 Ein Beispiel

Das folgende Beispiel, bekannt unter “Lea Sombé “ wird in [8] erstmals vorgestellt. Es wird im Folgenden um weitere Modellvariable und Zeitkonditionale erweitert. Über einen Zeitraum von $-2, -1, 0, 1, 2$ betrachten wir in einer fiktiven Population folgende Eigenschaften:

- $J = j, n$: Jung sein: ja, nein,
- $S = j, n$: Student sein: ja, nein,
- $F = s, v, e$: Familienstand: single, verheiratet, eheähnliches Verhältnis,
- $E = j, n$: Elternteil sein: ja, nein.

Desweiteren seien *LEA* und *COND* zwei ausgezeichnete Individuen in dieser Gesellschaft.

Zum Zeitpunkt -2 gelten folgende Sachzusammenhänge : 90 % aller Studenten sind jung, 30 % junger Menschen studieren, *LEA* ist Single. In der hier verwendeten Symbolik:

$$\mathcal{R}_{-2} = \{J = j | S = j [0.9], S = j | J = j [0.3], F = s | LEA [1.0]\}.$$

Weitere Sachzusammenhänge mögen zu den verschiedenen Zeitpunkten bekannt werden, sie sind nachstehend aufgeführt.

$$\mathcal{R}_{-1} = \{F = s | J = j [0.8], J = j | F = s [0.7], J = j | F = e [0.8], E = j | LEA [1.0]\}.$$

$$\mathcal{R}_0 = \{F = e \vee F = v | S = j \wedge E = j [0.9]\}.$$

$$\mathcal{R}_1 = \{E = j | J = j \wedge S = j [0.01], F = v | COND [1.0]\}.$$

$$\mathcal{R}_2 = \{S = j | E = n \wedge J = j [0.95], E = j | COND [1.0]\}.$$

Mit diesen Sachverhalten zuzüglich der Regeln \mathcal{R}_c für den richtigen chronologischen Zeitablauf lösen wir Aufgabe (3) und erhalten so eine Wahrscheinlichkeitsverteilung als Wissensrepräsentanten, aus der sich das bis zu den einzelnen Zeitpunkten erworbene Wissen durch Aktivierung und Ausblendung entsprechender (Zeit-)Variabler W_τ ableiten lässt.

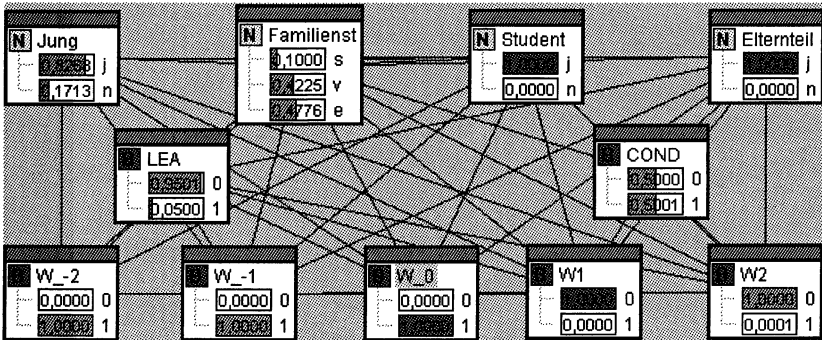


Abbildung 2. Gegenwärtiger Wissensstand.

- a) Wir wollen Wissen zum Zeitpunkt 0 abfragen. Dazu wird die Zeitvariable W_0 aktiviert, was unmittelbar auch die Aktivierung von W_{-2} und W_{-1} mit sich zieht; W_1 wird ausgeblendet und somit auch W_2 . Damit erfahren wir, dass zum gegenwärtigen Zeitpunkt in der Gesamtpopulation ca. 83 % aller Studenten mit Kindern jung sind ($P_0^*(J = j | S = j \wedge E = j) = 0.83$). Das zeigt der Bildschirmausdruck aus SPIRIT in Abbildung 2. Diese Aussage ändert sich jedoch signifikant, wenn zukünftige Entwicklungen mit berücksichtigt werden. So werden zum Zeitpunkt 1 wohl nur noch knapp

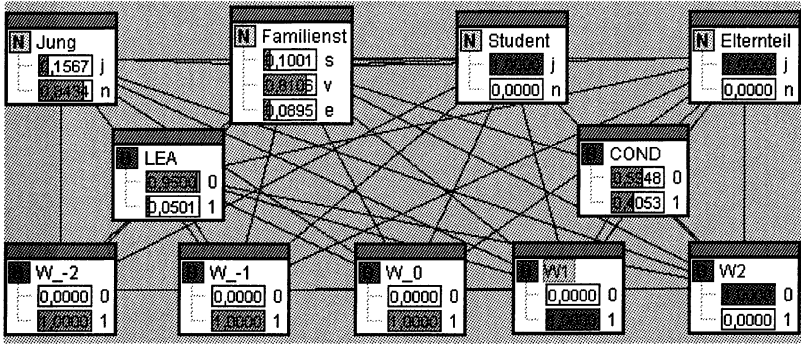
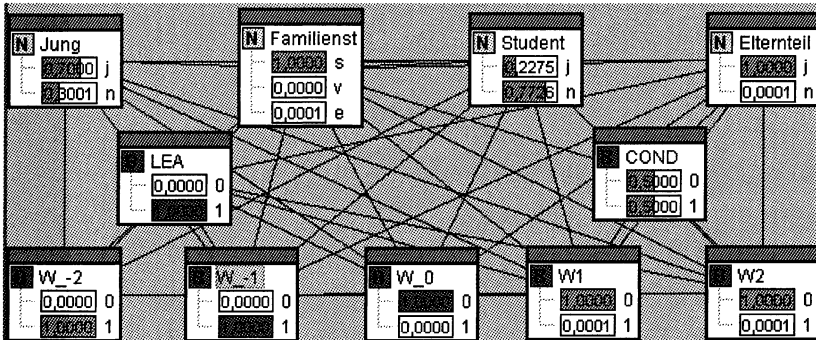


Abbildung 3. Zukünftiger Wissenstand.

16 % aller Studenten, die Kinder haben, jung sein ($P_1^*(J = j | S = j \wedge E = j) = 0.16$), vgl. Abbildung 3, will man der vermeintlich zukünftig zufließenden Information Glauben schenken.

Während hier Zusammenhänge über Eigenschaften nicht näher in der Population spezifizierter Individuen abgefragt wurden, betrachten wir im Folgenden zwei Individuen der Gesellschaft, *LEA* und *COND*.

- b) So erfahren wir aus Abbildung 4: $P_{-1}^*(F = s | LEA) = 1$, was bedeutet, dass *LEA* zum Zeitpunkt -1 Single war.

Abbildung 4. Wissensstand zum Zeitpunkt -1 .

- c) Weil $P_{-2}^*(E=j | LEA) = 0$, $P_{-1}^*(E=j | LEA) = 1$ und $P_0^*(E=j | LEA) = 1$ sind, vgl. Abbildung 4 und 5, ist das gleichbedeutend mit der Aussage:
LEA ist nicht immer Mutter gewesen.

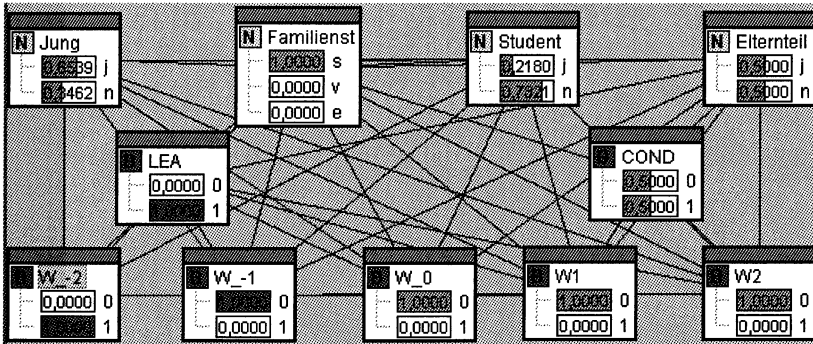


Abbildung 5. Wissensstand zum Zeitpunkt -2.

- d) Aus $P_{-2}^*(F=s \mid LEA) = 1$ und $P_{-1}^*(F=s \mid LEA) = 1$ leiten wir ab:
LEA war schon bis Zeitpunkt -1 stets Single gewesen.
- e) Aus Abbildung 5 erfahren wir, dass *LEA* zum Zeitpunkt -2 Single gewesen war aber noch keine Mutter.
 Es ist nämlich $P_{-2}^*(F=s \mid LEA) = 1$ und $P_{-2}^*(E=j \mid LEA) = 0$. Der Bildschirmausdruck in Abbildung 4 informiert darüberhinaus, dass *LEA* in -1 Mutter wurde: $P_{-1}^*(E=j \mid LEA) = 1$. Folglich erfahren wir aus der Kombination der beiden Wissensstände aus der Vergangenheit:
Bevor LEA Mutter wurde, war sie Single gewesen.

Ohne die entsprechenden Bildschirmausdrucke zu zeigen, berichten wir darüber, dass

- f) $P_0^*(F=v \mid COND) = 0$ und $P_1^*(F=v \mid COND) = 1$, was gleichbedeutend ist mit
COND wird (in 1) heiraten.
- g) $P_1^*(F=v \mid COND) = 1$ und $P_2^*(E=j \mid COND) = 1$, in Worten:
Wenn Cond Vater werden wird, wird er geheiratet haben.

Ohne einen allzu strengen Bezug zur Grammatik menschlicher Sprache herstellen zu wollen, müssen wir der vorgestellten Methode zur Bildung von Zeitformen jedoch eine große Aussagekraft bescheinigen. So kann man Sachverhalte im Jetzt (Präsens), in der Vergangenheit (Präteritum), in der bis ins Jetzt reichenden Vergangenheit (Perfekt) und schließlich auch in Bezug vor- und nachgelagerter Vergangenheitszeitformen (Plusquamperfekt) darstellen. Ähnliches gilt für zukünftige (Futur I) und relativ zukünftige Sachverhalte (Futur II). Der Leser mag anhand der Beispiele a) bis g) diese Aussagen überprüfen.

5 Fazit

In diesem Beitrag wurde gezeigt, wie der axiomatisch fundierte Max-Ent-Inferenzprozess um temporale Aspekte ergänzt werden kann. Während in der klassischen temporalen Logik neben den Konnektiven “und“, “oder“ sowie “nicht“ zum Ausdruck temporaler Sachzusammenhänge auf weitere Konnektive wie “immer“, “bevor“ etc. zurückgegriffen werden muss, geschieht die temporale Erweiterung in dem hier vorgestellten Ansatz über die Verwendung von Zeitkonditionalen. Damit ist es möglich, auf Basis des informationstreu- en Max-Ent-Prinzips über einen zeitlichen Modellhorizont hinweg Wissen, das bis zu einem bestimmten Zeitpunkt erworben wurde oder wird, jederzeit abzurufen. Auf diese Weise gestattet die Verwendung von Zeitkonditionalen auch die Formulierung von Sachzusammenhängen in verschiedenen Zeitformen, wie sie in der Alltagssprache verwendet werden, was die Mächtigkeit des konditionalen Zeitformenmodells eindrucksvoll unterstreicht.

Literatur

1. Calabrese, P. M. (1991) Deduction and Inference Using Conditional Logic and Probability, Conditional Logic in Expert Systems.I. R. Goodman, M. M. Gupta, H. T. Nguyen, G. S. Rogers (editors), Elsevier Science Publishers B. V.
2. Kern-Isberner, G. (2001) Conditionals in Nonmonotonic Reasoning and Belief Revision, Springer Berlin Heidelberg.
3. Kröger, F. (1987) Temporal Logic of Programs, Springer Berlin Heidelberg.
4. Puppe, F. (1993) Systematical Introduction to Expert Systems, Springer Berlin Heidelberg.
5. Reucher, E., Rödder, W. (2002) Wissensrevision in einer MaxEnt/MinRel-Umgebung, OR Proceedings, Springer Berlin Heidelberg 533-538.
6. Rödder, W. (2000) Conditional Logic and the Principle of Entropie, Artificial Intelligence, 117, 83-106.
7. Shore, J.-E., Johnson R.-W. (1980) Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy, IEEE Transact Inf Theory IT-26(1): 26-37.
8. Sombé, L. (1992) Schließen bei unsicherem Wissen in der künstlichen Intelligenz, Vieweg, Braunschweig, Wiesbaden.
9. SPIRIT (2002) Sprit-Version 3.1, <http://www.xspirit.de>.
10. Stirling, C. (1987) Comparing Branching Time Temporal Logics, Temporal Logic in Specification, Banieqbal B., Barringer, H., Pnueli, A. (editors), Springer Berlin Heidelberg.

Dienstplanbewertung mit unscharfen Regeln

Alexandra Schroll and Thomas Spengler

Fakultät für Wirtschaftswissenschaft, Otto-von-Guericke Universität Magdeburg

Zusammenfassung Die im vorliegenden Beitrag angestellten Überlegungen konzentrieren sich auf die Entwicklung eines unscharfen regelbasierten Systems zur automatisierten Bewertung von Dienstplänen. Als Inputgrößen des Systems modellieren wir die linguistischen Variablen Personalbedarfsdeckung und Mitarbeiterzufriedenheit mit ihren korrespondierenden linguistischen Termen. Die resultierende Outputgröße ist die Gesamtbewertung des Dienstplanes. Ein weiteres Augenmerk gilt neben der Modellierung des unscharfen Regelsystems der Lösung des in diesem Zusammenhang stehenden Skalierungsproblems.

1 Einleitung

In der Praxis der Dienstplanung wird vielfach die Forderung nach der Generierung guter Dienstpläne laut. Als gut werden solche Dienstpläne bezeichnet, die zum einen einen Beitrag zur Erfüllung der Unternehmensziele leisten und zum anderen auf Akzeptanz der betroffenen Mitarbeiter stoßen (vgl. [6], [11]). Sie führen u. a. zu möglichst geringen Abweichungen vom Personalbedarf (Perspektive des Unternehmens) und sehen z. B. - wenn möglich - die Gewährung geblockter freier Tage vor (Perspektive der Mitarbeiter). Da die zu erfüllenden Kriterien meist in unscharfer Form vorliegen und diese Unschärfe mit traditionellen Verfahren nicht adäquat verarbeitet werden kann, bietet es sich an, auf Fuzzy-Konzepte (insb. auf Verfahren des Fuzzy-Control) zurückzugreifen (vgl. [1], [4], [7]). Darüber hinaus ist zu beachten, dass die verwendeten Input- und Outputgrößen hinsichtlich ihrer Bewertung in hohem Maße durch Subjektivismen geprägt sind. Dies zeigt sich beispielsweise in der unterschiedlichen Wahrnehmung und Beurteilung von Schichtmustern, aber auch bei der Evaluation von Personalbedarfsabweichungen. Dienstpläne werden i. d. R. von einer einzelnen Person, dem sog. Dienstplaner erstellt. Seine Aufgabe ist es, nach eigenem Ermessen gute Dienstpläne zu erstellen und zwar unter Berücksichtigung der Perspektive des Unternehmens und der Perspektive der Mitarbeiter: Dies impliziert letztendlich, dass die Bewertung von subjektiven Einschätzungen des Dienstplaners abhängt, der mitunter jedoch die Präferenzen der Mitarbeiter antizipiert.

Da bereits bei relativ einfachen Problemstellungen eine Vielzahl theoretisch möglicher und zulässiger Dienstpläne existiert, ist eine systematisch-methodische sowie automatisierte Generierung und Evaluierung von Dienstplänen erforderlich (vgl. [8], [10], [11]). Im folgenden präsentieren wir ein Modell, welches die subjektiven Vorstellungen des Dienstplaners hinsichtlich der

Güte von Personalbedarfsdeckungen und der Güte von Schichtmustern formal abbildet und diese im Rahmen eines unscharfen regelbasierten Systems verarbeitet. Als Ergebnis erhalten wir die Gesamtbewertung von Dienstplänen aus Unternehmens- und Mitarbeitersicht.

2 Modellannahmen

Bei dem zu konstruierenden Modell handelt es sich um einen bewusst einfach gehaltenen Ansatz zur Bewertung von Dienstplänen. Das Modell beruht auf den nachfolgend aufgeführten Annahmen:

- Wir betrachten aus dem Kanon der Dienstplanungsansätze ein Days-Off-Planungsproblem, d. h. wir bewerten Dienstpläne bei denen Mitarbeitern arbeitsfreie Tage und Arbeitstage zugeordnet werden (vgl. [8],[10]). Der Planungszeitraum beträgt eine Woche, wobei wir die Tage mit $t=\text{Mo., Di., ..., So.}$ bezeichnen.
- Ziel des Unternehmens ist es, den Personalbedarf möglichst exakt zu decken. Folglich sind Personalbedarfsunterdeckungen und -überdeckungen zulässig. Ein Dienstplan erfährt jedoch eine um so schlechtere Bewertung, je größer die Differenz zwischen Ausstattung und Bedarf ausfällt. Wir bewerten hierbei das arithmetische Mittel aller Abweichungen einer Woche, wobei zur Berechnung der Abweichungen die Formel $\frac{|PA_t - PB_t|}{PB_t}$ berücksichtigt wird. Dies impliziert, dass wir Personalbedarfsunterdeckungen und -überdeckungen als gleichwertig einstufen, die Abweichungsbewertung jedoch von der absoluten Höhe des Personalbedarfs abhängt. In Tabelle 2.2 ist exemplarisch ein Dienstplan dargestellt. Die relativen täglichen Personalbedarfsabweichungen finden sich in der letzten Zeile, die aggregierte Abweichung beträgt 0,079. Diese Größe gilt es, im Rahmen des sich in Kapitel 3 anschließenden Modells zu bewerten.
- Das Unternehmen bietet den Arbeitskräften ausschließlich Schichtmuster des Typs 5+2 an (5 Arbeitstage und 2 freie Tage). Bei den Schichtmustern unterscheiden wir zum einen, ob konsekutive oder nicht konsekutive freie Tage gewährt werden und zum anderen, welche Tage (Wochentag oder Wochenende) als freie Tage vorgesehen sind. Nach dieser Differenzierung erhalten wir fünf Schichtgrundmuster (G_i) (s. Table 1).
- Jede Arbeitskraft kann prinzipiell in jeder Schicht bzw. an jedem Tag zur Personalbedarfsdeckung herangezogen werden. Darüber hinaus verbinden Arbeitskräfte mit den oben angegebenen Schichtgrundmustertypen unterschiedliche Nutzenwerte $U(G_i)$. Wir berücksichtigen hierbei die Nutzenwerte eines repräsentativen Durchschnittsmitarbeiters (s. Table 1).
- Der Nutzen bzw. die Zufriedenheit der Mitarbeiter mit einem Dienstplan hängt von dem jeweils zugeteilten Schichtmuster ab. Dabei gehen wir davon aus, dass der Mitarbeiter bei der Beurteilung des Dienstplanes nur den Dienstplan an sich und nicht das ihm zugewiesene Schichtmuster

Tabelle1. aktive Regeln

konsekutiv			nicht konsekutiv	
Wochenende	Wochenendtag/ Wochentag	Wochentag	Wochentag/ Wochenendtag	Wochentage
G_1	G_2	G_3	G_4	G_5
$U(G_1)=6$	$U(G_2)=5, 1$	$U(G_3)=4, 4$	$U(G_4)=2$	$U(G_5)=0, 7$

Tabelle2. Dienstplan

t	Mo.	Di.	Mi.	Do.	Fr.	Sa.	So.	$U(G_i)$
PB_t	17	12	15	21	14	16	8	
	6	-	-	6	6	6	6	4,4
	2	2	2	-	-	2	2	4,4
	3	3	3	3	-	-	3	5,1
	2	2	2	2	2	-	-	6
	-	4	4	4	4	4	-	5,1
	2	-	2	2	2	2	-	2
	2	2	2	2	-	2	-	2
PA_t	17	13	15	19	14	16	11	
$\frac{ PA_t - PB_t }{PB_t}$	0	0,083	0	0,095	0	0	0,375	

kennt. Er ist jedoch in der Lage, sich über den Erhalt eines Schichtmusters ein Wahrscheinlichkeitsurteil zu bilden. Als Kalkül für die Präferenz fungiert der Erwartungswert des Nutzens eines Dienstplanes ($EU(G)$); dieser resultiert aus der Wahrscheinlichkeit des Erhalts eines Schichtmusters vom Grundtyp i (p_i) multipliziert mit dem Nutzen eines Schichtmusters vom Grundtyp i ($U(G_i)$) summiert über alle i . Formal dargestellt gilt der folgende Zusammenhang (vgl.[3]):

$$EU(G) = \sum_{i=1}^5 p_i * U(G_i) \quad (1)$$

Für den in Table 2 dargestellten Dienstplan, ergibt sich gemäß der oben aufgeführten Formel für den Erwartungswert des Nutzens ein Wert in Höhe von 4,33. Diesen bewerten wir im nachfolgenden Modell, indem wir Zugehörigkeitswerte zur Menge der sehr guten bis sehr schlechten Dienstpläne zuordnen.

- Die Gesamtbewertung eines Dienstplanes hängt von der Güte der Personalbedarfsdeckung und von der Güte der Schichtmuster ab (vgl. [11]).

3 Modellformulierung

3.1 Modellierung der linguistischen Variablen

Im ersten Schritt wollen wir uns nun der Bewertung der aggregierten relativen Abweichungen vom Personalbedarf widmen. Dazu modellieren wir für die linguistische (Input-)Variable Personalbedarfsdeckung fünf linguistische Terme (sehr gute, gute, mittlere, schlechte und sehr schlechte Personalbedarfsdeckung) und bilden diese auf der Grundmenge der aggregierten relativen Abweichungen vom Personalbedarf ab. Die subjektiv festgelegten Zugehörigkeitsfunktionen der linguistischen Terme zeigt Fig. 1 (vgl. [12]).

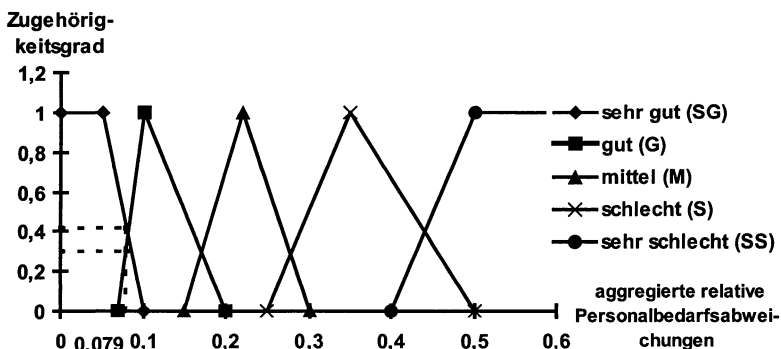


Abbildung1. linguistische Variable Personalbedarfsdeckung (PB)

Die Bewertung der Personalbedarfsdeckung im obigen Beispiel erfolgt nunmehr in fuzzyfizierter Form. Mit einem Zugehörigkeitswert von 0,3 gehört die Abweichung von 0,079 zur Menge der guten Personalbedarfsdeckungen und mit einem Zugehörigkeitswert von 0,42 zur Menge der sehr guten Personalbedarfsdeckungen. Diese Ergebnisse sind auch aus Fig. 1 ablesbar.

Die linguistische (Input-)Variable Mitarbeiterzufriedenheit charakterisieren wir ebenfalls anhand fünf linguistischer Terme (sehr hoch, hoch, mittel, niedrig, sehr niedrig) und bilden diese auf der Grundmenge Erwartungsnutzen ab. Für die scharfe Inputgröße des obigen Beispiels in Höhe von 4,33 erhalten wir einen Zugehörigkeitswert von 0,07 zum linguistischen Term mittlere Zufriedenheit und mit 0,8 zum linguistischen Term hohe Zufriedenheit (s. Fig. 2).

Als letzte Größe verbleibt nun die Modellierung der linguistischen (Output-)Variable Gesamtbewertung Dienstplan. Wir charakterisieren diese anhand fünf linguistischer Terme, die wiederum mit sehr gut, gut, mittel,

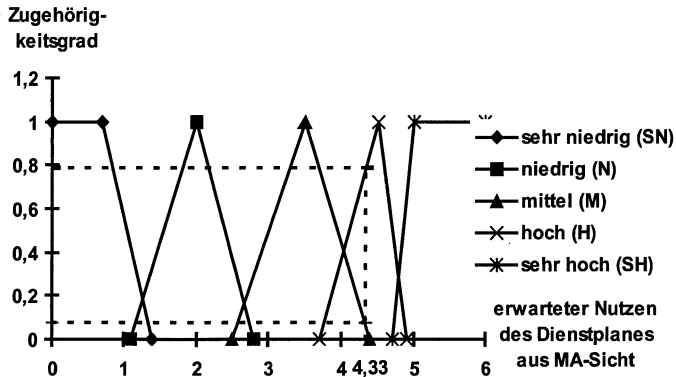


Abbildung 2. linguistische Variable Mitarbeiterzufriedenheit (MZ)

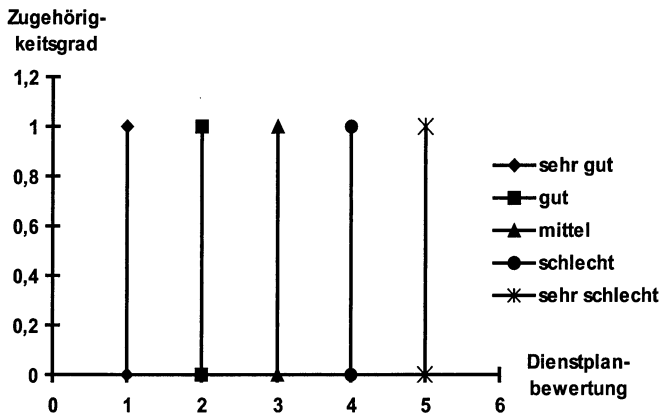


Abbildung 3. linguistische Variable Gesamtbewertung Dienstplan (D)

schlecht und sehr schlecht bezeichnet werden. Im Gegensatz zu den beiden linguistischen Variablen Personalbedarfsdeckung und Mitarbeiterzufriedenheit, liegt uns jetzt eine ordinal skalierte Grundmenge in Form einer Notenskala von 1 bis 5 vor (beste Note = 1, schlechteste Note = 5). Für die Modellierung hat dies zur Konsequenz, dass wir die linguistischen Terme nur in Form von Singletons abbilden können (s. Fig. 3).

3.2 Formulierung der zugrundegelegten Regelbasis und Ermittlung der resultierenden Outputmenge

Die Verknüpfung der linguistischen In- und Outputgrößen realisieren wir über eine Regelbasis, die (hier) aus 25 Regeln besteht und sich aus der Kombi-

nation aller linguistischen Terme der linguistischen Variable Mitarbeiterzufriedenheit (MZ) mit den linguistischen Termen der linguistischen Variable Personalbedarfsdeckung (PB) ergibt. Die mit den jeweiligen Regeln korrespondierenden Schlussfolgerungen liegen im Ermessen des Dienstplaners: Er entscheidet bei Vorliegen bestimmter Prämissen aus seiner Erfahrung heraus über die Einstufung eines Dienstplanes. In Table 3 sind exemplarisch vier Regeln dargestellt.

Tabelle3. aktive Regeln

R_3	WENN	PB SG und MZ M	DANN	D G
R_4	WENN	PB SG und MZ H	DANN	D SG
R_8	WENN	PB G und MZ M	DANN	D G
R_9	WENN	PB G und MZ H	DANN	D G

Die weitere Vorgehensweise beschreiben wir wiederum unter Bezugnahme auf das in Kapitel 2 eingeführte Beispiel. Die dort ermittelten scharfen Input-Werte liegen bereits in fuzzyfzierter Form vor. Der nächste Schritt der Inferenz besteht nun darin, die aktiven Regeln zu identifizieren, wobei eine Regel dann aktiv ist, wenn deren sämtliche Prämissen erfüllt sind (z. B. Zugehörigkeit zur Menge der PB SG und Zugehörigkeit zur Menge MZ M > 0). Bezogen auf unser bisheriges Beispiel, existieren dann vier aktive Regeln (R_3 , R_4 , R_8 und R_9) und zwar diejenigen, die zu den Prämissen PB = sehr gut und MZ = hoch, PB = sehr gut und MZ = mittel, PB = gut und MZ = hoch sowie PB = gut und MZ = mittel gehören (s. Table 3).

Zur Ermittlung der einzelnen Output-Fuzzy-Mengen müssen wir zunächst die Erfüllungsgrade (H_i) der aktiven Regeln ermitteln. Da die Prämissen der Regeln UND-verknüpft sind, wählen wir zur Verknüpfung der Zugehörigkeitswerte aus der Gruppe der t-Normen den Minimumoperator (vgl. [1], [2], [7]). Als Ergebnis erhalten wir für unsere aktiven Regeln: $H_3=0,07$, $H_4=0,42$, $H_8=0,07$, $H_9=0,3$, wobei sich z. B. H_3 aus der Operation $\text{Min}(0,42; 0,07) = 0,07$ ergibt.

Die zu gleichen Konsequenzen führenden Regeln (im vorliegenden Beispiel R_3 , R_8 und R_9) überlagern wir dann mit dem Maximumoperator, kappen die Fuzzy-Mengen der Schlussfolgerung jeder aktiven Regel in der Höhe des jeweiligen Erfüllungsgrades und tragen diese auf der Outputgröße Gesamtbewertung Dienstplan ab. Durch Überlagerung der einzelnen Fuzzy-Mengen mit dem Maximumoperator erhalten wir die in Fig. 4 dargestellte resultierende Output-Fuzzy-Menge (vgl. [4], [5], [7], [9]). Diese beinhaltet die Bewertung des vorliegenden Dienstplanes als sehr gut mit dem Zugehörigkeitswert 0,42 und als gut mit dem Zugehörigkeitswert 0,3.

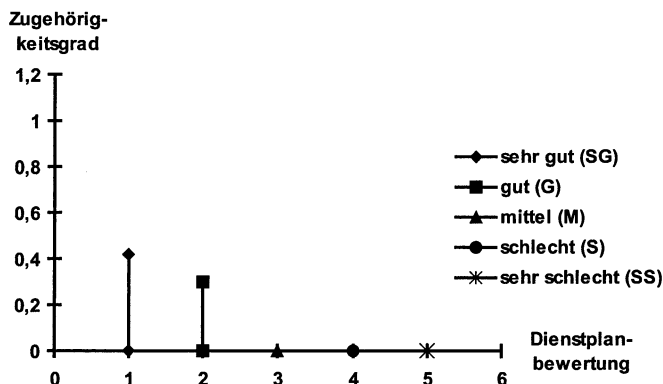


Abbildung 4. resultierende Outputmenge

3.3 Defuzzifizierung der unscharfen Outputmenge

Der letzte Schritt unseres Modellansatzes besteht nun darin, eine konkrete Handlungsempfehlung bzw. eine abschließende Bewertung des Dienstplanes vorzunehmen (vgl. zu Defuzzifizierungsverfahren [4], [5]). Da die linguistischen Terme unserer Outputvariable in Form von Singletons dargestellt werden, ist unseres Erachtens die Maximummethode, wonach die Regel mit dem höchsten Erfüllungsgrad (R_4) den Ausschlag bei der Bewertung gibt, das geeignetste Verfahren. Der im vorliegenden Beispiel verwendete Dienstplan ist demnach in die Kategorie der sehr guten Pläne einzuordnen.

4 Schlussbemerkungen

Der vorliegende Beitrag erläutert die Konzeption eines Modells zur Bewertung von Dienstplänen auf Basis unscharfer regelbasierter Systeme. Als Inputgrößen dieses Modells dienen die linguistische Variable Personalbedarfsdeckung und die linguistische Variable Mitarbeiterzufriedenheit. Die Outputgröße stellt die linguistische Variable Gesamtbewertung Dienstplan dar. Besonderes Charakteristikum des Modells ist das unterschiedliche Skalenniveau der In- und Outputgrößen. Während die Inputgrößen auf einer metrischen Skala gemessen werden, weist die Outputgröße eine ordinale Skalierung auf. Im Rahmen der Modellformulierung zeigen wir, wie sowohl ordinale als auch metrisch skalierte Größen in unscharfen regelbasierten Systemen formal dargestellt und verarbeitet werden können.

Zukünftige Forschungsarbeiten werden sich mit der Erweiterung dieses Modellansatzes zu beschäftigen haben. Dabei wird es vor allem um die Erweiterung und Modifikation der Modellannahmen gehen. Mitunter sollen weitere Schichtmustertypen integriert, die Bewertung der Personalbedarfsabweichungen modifiziert und weitere Arbeitskräftekategorien berücksichtigt werden.

Literatur

1. Bellmann R E, Zadeh L (1970) Decision Making in a Fuzzy Environment. *Management Science*, Vol 17: B-141-B-164
2. Calvo T, Kolesarova A, Kormonikova M, Mesiar R (2003) Aggregation Operators: Properties, Classes and Construction Methods. In: Calvo T, Mayor G, Mesiar R (Hrsg) *Aggregation Operators. New Trends an Applications*, Physica, Heidelberg, New York, S. 3-104
3. Eisenführ F, Weber M (2003) *Rationales Entscheiden*. 4 Aufl, Springer, Berlin, Heidelberg, New York, S. 3-104
4. Jaanineh G, Maijohann M (1996) *Fuzzy-Logik und Fuzzy-Control*. Vogel, Würzburg
5. Kahlert J, Frank H (1993) *Fuzzy-Logik und Fuzzy-Control. Eine anwendungsorientierte Einführung*. Vieweg, Braunschweig, Wiesbaden
6. Kieper F, Spengler T (2002) Das 3-Säulenmanagement und Fuzzy-Control. *Der Controlling-Berater* H 3: 69-88
7. Kruse R, Gebhardt J, Klawonn F (1992) *Fuzzy-Systeme*. B. G. Teubner, Stuttgart
8. Nanda R, Browne J (1992) *Introduction to Employee Scheduling*. Von Nostrand Reinhold, New York
9. Rommelfanger H (1994) *Fuzzy Decision Support-Systeme. Entscheiden bei Unschärfe*. 2. Aufl, Springer, Heidelberg
10. Salewski F (1998) Klassifikation von Dienstplanungsproblemen. In: Kossbiel H (Hrsg) *Modellgestützte Personalentscheidungen 2*. Rainer Hampp, München, Mering S 119-136
11. Schroll A, Spengler T (2002) Fuzzy-Control in der Personaleinsatzplanung. In: Kossbiel H, Spengler T (Hrsg) *Modellgestützte Personalentscheidungen 6*. Rainer Hampp, München, Mering S 121-140
12. Zadeh L (1975) The Concept of a Linguistic Variable and Its Application to Approximate Reasoning. Part 1. *Information Sciences* 8: 199-249

Optimization by Gaussian Processes assisted Evolution Strategies

Holger Ulmer, Felix Streichert, and Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72074 Tübingen, Germany
ulmerh,streiche,zell@informatik.uni-tuebingen.de,
<http://www-ra.informatik.uni-tuebingen.de>

Abstract. Evolutionary Algorithms (EA) are excellent optimization tools for complex high-dimensional multimodal problems. However, they require a very large number of problem function evaluations. In many engineering optimization problems, like high throughput material science or design optimization, a single fitness evaluation is very expensive or time consuming. Therefore, standard evolutionary computation methods are not practical for such applications. Applying models as a surrogate of the real fitness function is a quite popular approach to handle this restriction. We propose a Model Assisted Evolution Strategy (MAES), which uses a Gaussian Process (GP) approximation model. The purpose of the Gaussian Process model is to preselect the most promising solutions, which are then actually evaluated by the real problem function. To refine the preselection process the likelihood of each individual to improve the overall best found solution is determined. Numerical results from extensive simulations on high dimensional test functions and one material optimization problem are presented. MAES has a much better convergence rate and achieves better results than standard evolutionary optimization approaches with less fitness evaluations.

1 Introduction

Evolution Strategies (ES) are one class of Evolutionary Algorithms (EAs) which are often used as optimization tools for complex high dimensional multimodal problems [8] [9]. In contrast to other EAs like Genetic Algorithms or Genetic Programming ES work directly on real valued objective variables, which represent a possible solution. Therefore ES are very suitable for many engineering optimization problems.

However, like other population based EAs ES require a very high number of fitness function evaluations to determine an acceptable solution. In most real world engineering optimization applications the process of fitness evaluation is very expensive and time consuming. Therefore standard ES methods are not practical for such applications.

A promising approach to handle this problem is the application of modeling techniques, where a model evaluation is orders of magnitude cheaper than

a real fitness function evaluation. A model is trained on already evaluated fitness cases and is used to guide the search for promising solutions. This approach decreases the number of expensive fitness evaluations and has a better convergence rate. The application of modeling techniques in evolutionary computation receives increasing attention [7] [2] [3] [10]. A survey on this research field can be found in [6].

The remainder of this paper is organized as follows: Section 2 introduces the synthesis of the Gaussian Process (GP) fitness approximation model with a standard ES. The GP model is utilized to assist the ES by selecting the most promising solutions of an offspring population to be evaluated by the real fitness function. Numerical results from extensive simulations on high dimensional artificial test functions and one material optimization task are presented and discussed in section 3. The paper closes with a brief conclusion and outlook on future work.

2 GP Model Assisted Evolution Strategy

For the approximation of the fitness function we chose Gaussian Processes (GP), which are general and proper real valued function approximators, especially for noisy training data. A detailed description is given in [4]. Compared to other models like artificial neural networks GP are probabilistic models, which have the advantage of providing a confidence value given by the standard deviation σ for the predicted fitness value t without additional computational cost. Moreover GP are stable against overfitting and have only a limited number of model parameters, which have to be chosen by the user. [9]. We start our consideration with a standard (μ, λ) ES, which will be later coupled with the GP model. An ES works on a population of potential solutions \mathbf{x} (individuals) by manipulating these individuals with evolutionary operators [8]. By applying the evolutionary operators reproduction, recombination and mutation (see pseudocode in Figure 1) λ offspring individuals are generated from μ parents. After evaluating the fitness of the λ offspring individuals, μ individuals with the best fitness are selected by a (μ, λ) strategy to build the parent population for the next generation. The algorithm terminates when a maximum number of fitness function evaluations have been performed.

To incorporate the approximation model into the ES we use a pre-selection concept similar to the one described by Emmerich et al. [3]. Compared to the standard ES $\lambda_{Pre} > \lambda$ new offspring individuals are created from μ parents (see pseudocode in Figure 2). These λ_{Pre} individuals have to be pre-selected to generate the offspring of λ individuals, which will be evaluated with the real fitness function. The model is trained at the beginning with a randomly created initial population and is updated after each generation step with λ new fitness cases. The key point of our approach is the pre-selection procedure. Using the mean of the model prediction to identify the most promising

Procedure ES

```

Begin
  eval=0;
  Pop=CreateInitialPop();
  Pop.EvaluateRealFitness();

  while (eval<maxeval);
    Offspring=Pop.Reproduce( $\lambda$ );
    Offspring.Mutate();

    Offspring.EvaluateRealFitness();

    Pop=Offspring.SelectBest( $\mu$ );
    eval=eval+ $\lambda$ ;
  end while
End

```

Fig. 1. Standard (μ, λ) Evolution Strategy (ES).

Procedure MAES

```

Begin
  eval=0;
  Pop=CreateInitialPop();
  Pop.EvaluateRealFitness();
  Model.update(Pop);
  while (eval<maxeval);
    PrePop=Pop.Reproduce( $\lambda_{Pre}$ );
    PrePop.Mutate();
    PrePop.EvaluateWithModel();
    Offspring=PrePop.SelectBest( $\lambda$ );
    Offspring.EvaluateRealFitness();
    Model.update(Offspring);
    Pop=Offspring.SelectBest( $\mu$ );
    eval=eval+ $\lambda$ ;
  end while
End

```

Fig. 2. Model Assisted Evolution Strategy (MAES).

individuals leads to premature and suboptimal convergence rate on multi-modal problems with many misleading local minima, because individuals with a better model prediction are preferred to others and therefore have a lower probability to escape from these minima.

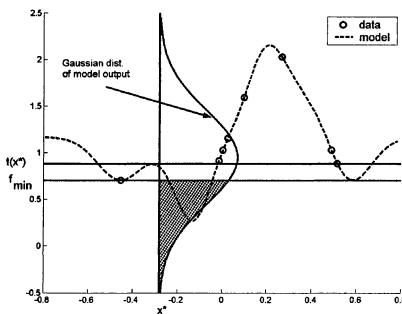


Fig. 3. The gray filled area represents the probability that a model output value t is sampled at point x^* , which is smaller than f_{min} (POI).

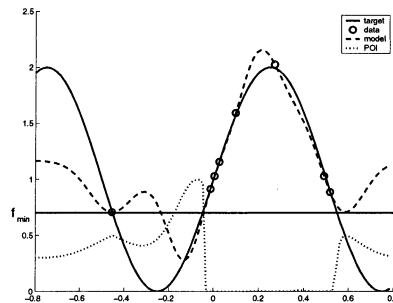


Fig. 4. Areas with a higher POI criterion have a higher probability to sample a data point with a target value smaller than f_{min} .

To address this problem we use a new pre-selection criterion, which utilizes the model confidence given by the GP model as the standard deviation $\sigma(x)$.

The idea is not new in the field of global optimization [1], but new in the context of evolutionary optimization. The concept is illustrated in Figure 3. At any given point \mathbf{x} , we model the uncertainty about the model value prediction by considering this value to be like the realization of a random variable $Y(\mathbf{x})$ with mean $\hat{t}(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$.

Let $f_{min} = \min(t_1, \dots, t_N)$ be the current best fitness value sampled until now, then the target value for the improvement will be some number $T \leq f_{min}$. The Probability Of Improvement (POI) is simply the probability that $Y(\mathbf{x}) \leq T$. Assuming the random variable is normal distributed, this probability is given by

$$POI(\mathbf{x}) = \Phi\left(\frac{T - \hat{t}(\mathbf{x})}{\sigma(\mathbf{x})}\right) \quad (1)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function. Figure 4 shows the characteristics of the POI selection criterion. Areas with a high POI have a high probability to sample a data point with a target value smaller than f_{min} and are therefore more promising. Areas with model prediction $\hat{t}(\mathbf{x}) \gg f_{min}$ have a low probability of improvement $POI \approx 0$. As the function is sampled more and more around the current best point, the standard deviation in this area decreases. The term $\frac{T - \hat{t}(\mathbf{x})}{\sigma(\mathbf{x})}$ becomes extremely negative and POI will be so small that the algorithm is driven to search elsewhere in unexplored areas where the standard deviation is higher. Therefore POI prefers unexplored areas of object space and has a multimodal characteristic. Note, that the maximal POI value may have another location in object space than the minimal model output value. The individuals $\mathbf{x}_i, i = 1, \dots, \lambda_{Pre}$ with the highest $POI(\mathbf{x})$ are pre-selected to build the new offspring.

The size of the pre-selected population λ_{Pre} controls the impact of the model on the evolutionary optimization process. For $\lambda_{Pre} = \lambda$, the algorithm performs like a standard (μ, λ) ES. Increasing λ_{Pre} results in a larger selection pressure in the pre-selection and in a stronger impact of the model on the convergence behavior of the optimization process.

3 Experimental Results and Discussion

To analyze the algorithms extensive numerical simulations were performed for 5 artificial test functions and one material optimization problem. For each case the standard (μ, λ) -ES algorithm is compared with the new (μ, λ) -MAES algorithm for population size $(\mu = 5, \lambda = 35)$.

The size of the pre-selected population λ_{Pre} was set to 3λ . We used Covariance Matrix Adaption (CMA) developed by Hansen et al.[5], which is a powerful method for adaption of the mutation step size. For all simulations no recombination was used and the initial population size was set to 10.

The training data for the GP model consisted of the 2λ most recently performed fitness evaluations. For this reason the model is a local model of the individual's neighborhood in object space. Using more training data improves the performance only slightly but results in with much higher computational costs for model training.

The values are always evaluated as the mean of 100 repeated runs with different seed values for random number generation.

The Sphere function $f_{Sphere}(\mathbf{x}) = \sum_{i=1}^{20} x_i^2$ is a nonlinear and unimodal test function which is a good test for the self-adaptation mechanism of ES.

The MAES (see Figure 5) shows a better convergence rate and outperforms the standard ES clearly. It reaches 10 times better fitness values after 5000 evaluations. Comparable results are obtained with several other unimodal functions. In Figure 6 results are presented for the Schwefel 1.2 test function:

$$f_{Schwefel}(\mathbf{x}) = \sum_{i=1}^{20} \left(\sum_{j=1}^i x_j \right)^2.$$

The Rosenbrock function (2) is nonlinear, continuous and not symmetric.

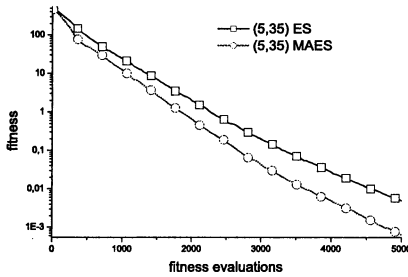


Fig. 5. 20-dim. Sphere function: fitness of best individual.

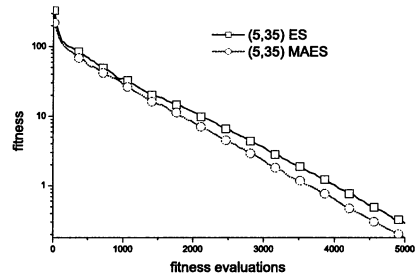


Fig. 6. 20-dim. Schwefel's function 1.2: fitness of best individual.

$$f_{Rosen}(\mathbf{x}) = \sum_{i=1}^{20} (100 \cdot (x_{i+1} - x_i)^2 + (x_i - 1)^2) \quad (2)$$

It is a very popular test function and has a very hard to find global optimum. Figure 7 shows that MAES reaches better fitness values than the standard ES. These results justifies the motivation of the model assisted approach to support the algorithm by identifying the most promising individuals. Multimodal functions evoke hills and valleys, which are misleading local optima. A simple optimization algorithm like hill-climbing would get stuck in a local minimum. For such problems evolutionary algorithms are much more appropriate.

Ackley's test function (3) is symmetric and very bumpy.

$$f_{Ack}(\mathbf{x}) = 20 + e - 20 \exp \left(-0.2 \cdot \sqrt{\frac{1}{20} \cdot \sum_{i=1}^{20} x_i^2} \right) - \exp \left(\frac{1}{20} \sum_{i=1}^{20} \cos(2\pi x_i) \right) \quad (3)$$

Its number of local minima increases exponentially with the problem dimension and has a global optimum with very strong local features. Here MAES converges faster than the standard ES (see Figure 8). Comparable results are obtained for Rastrigin's test function (3) (see Figure 9).

$$f_{Rast}(\mathbf{x}) = 10 \cdot 20 + \sum_{i=1}^{20} (x_i^2 - \cos(2\pi x_i)) \quad (4)$$

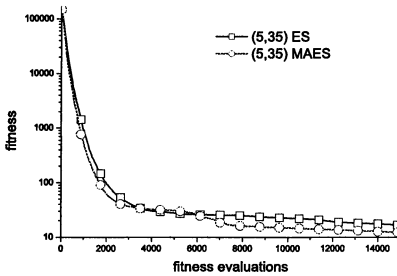


Fig. 7. 20-dim. Rosenbrock function: fitness of best individual.

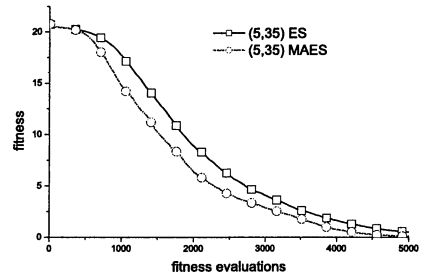


Fig. 8. 20-dim. Ackley's function: fitness of best individual.

The last presented result shows the application of the new MAES algorithm on a real world engineering optimization. Here the task is to optimize certain chemical catalytic properties of solid state samples created by a high throughput process. The objective function of the problem is modeled by a artificial neural network with data from already performed experiments and depends on 6 input variables, which describe the synthesis of the samples. The MAES algorithm outperforms the standard ES (see Figure 10). From the beginning MAES has a higher convergence rate and yields better solutions. It clearly yields better solutions with less fitness function evaluations. For example MAES reaches 2 times better solutions after 400 evaluations than the standard ES. Therefore MAES halves the costs of optimization for this application.

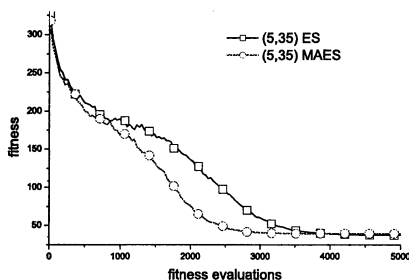


Fig. 9. 20-dim. Rastrigin's function: fitness of best individual.

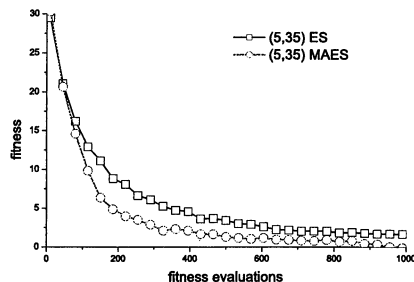


Fig. 10. 6-dim. Material optimization: fitness of best individual.

4 Conclusions

We applied a Gaussian Process as an approximation model to assist a standard ES by using the GP to pre-select the most promising individuals to be evaluated by the real fitness function.

The pre-selection procedure is given by the Probability Of Improvement (POI) pre-selection criterion. POI addresses the tradeoff between exploitation and exploration by utilizing the probabilistic interpretation of the GP model. This is done by evaluating the likelihood of each individual to improve the overall best solution.

Extensive simulations on artificial test functions showed that this approach enhances the performance of a standard ES on unimodal and multimodal problems. MAES has a higher convergence speed and is much more stable against premature convergence for multimodal problems. This is reasonable, because the MAES with POI pre-selection criterion has a higher tendency to sample in unexplored areas.

For a material optimization problem MAES yields the same solution quality with half the function evaluations and achieves overall better solutions. Therefore the application of MAES halves the costs compared to the standard ES method. These encouraging results justify the application of MAES in the field of engineering optimization applications where problem evaluations are very costly.

For further work it is planned to develop a mechanism which controls the impact of the approximation model on the optimization process by controlling λ_{Pre} . This can be carried out by using the confidence of the approximation model.

Acknowledgments This research has been funded by the German federal ministry of research and education (BMBF) in the project "Entwicklung eines Systems zur automatisierten Herstellung und Charakterisierung von kristallinen Festkörpern in hohem Durchsatz" under contract No. 03C0309E.

References

1. M. Schonlau D. Jones and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
2. M. A. El-Beltagy and A. J. Keane. Evolutionary optimization for computationally expensive problems using gaussian processes. In CSREA Press Hamid Arabnia, editor, *Proc. Int. Conf. on Artificial Intelligence IC-AI'2001*, pages 708–714, 2001.
3. M. Emmerich, A. Giotis, M. Multlu Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Parallel Problem Solving from Nature VII*, pages 362–370, 2002.
4. M. Gibbs and D. MacKay. Efficient implementation of gaussian processes. Technical report, Cavendish Laboratory, Cambridge, UK, 1997.
5. N. Hansen and A. Ostermeier. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_i, \lambda)$ -cma-es. In *5th European Congress on Intelligent Techniques and Soft Computing*, pages 650–654, 1997.
6. Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 2003. In press.
7. Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*. March 2002 (in press). (ISSN: 1089-778X), 2002.
8. I. Rechenberg. *Evolutionsstrategie '94*. frommann-holzboog, Stuttgart, 1994.
9. H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser, Basel, 1977.
10. H. Ulmer, F. Streichert, and A. Zell. Model-assisted steady-state evolution strategies. In *GECCO 2003 Proceedings of Genetic and Evolutionary Computation Conference*, pages 610–621, 2003.

Stochastic Programming and Statistical Estimates

Silvia Vogel

TU Ilmenau, Institut für Mathematik
Postfach 100565, 98684 Ilmenau
Silvia.Vogel@tu-ilmenau.de

Abstract. The paper aims at drawing attention to the potential of (qualitative) stability theory of stochastic programming for the study of asymptotic properties of statistical estimates. Making use of existing stability results, it is often possible to weaken the assumptions and to deal also with non-standard estimation problems. Thus non-unique solutions to the underlying optimization problems as well as constraints for the estimates can be taken into account and the continuity assumptions can often be replaced by semicontinuity. In this paper the focus is on consistency for constrained estimation with non-unique solutions. It will be shown how stability results obtained by the author can be employed to derived assertions on the convergence in probability of statistical estimates. The general results use the epi-convergence approach.

1 Introduction

Roughly spoken, stochastic programming problems deal with optimization problems which contain random elements in the objective functions and/or the constraints. Often they originate from decision problems where an optimal decision has to be chosen before random variables which influence the outcome are completely known. Then one usually looks for decisions optimizing some appropriate deterministic surrogate problem which usually contains expectations with respect to the distribution of the random variables involved ([23], [14]). In order to solve such a problem numerically, approximations, e.g. via estimates for unknown parameters or integration quadratures for the integrals, play an important role. Hence there has been a need for stability statements in order to judge whether solutions to the approximate problems come close to the solutions of the true problems. As the approximate problems are often based on estimates, convergence statements on the optimal values and the solution sets have been derived in the deterministic setting as well as in the ‘almost surely’ and the ‘in probability’ setting (cf. [19], [10] and the papers quoted there). The results apply to constrained problems and allow for non-unique solutions.

Stability statements for convergence ‘in distribution’ are also available ([21]). They are not of interest when a deterministic surrogate (decision) problem is approximated, but are promising when optimization problems

with respect to stochastic processes are investigated. They can also be employed when so-called distribution problems (cf. [14]) are studied.

The potential that stability theory of stochastic programming bears for asymptotic statistics was soon recognized ([2], [16]). Many estimators have proved to be solutions to random optimization problems. Consistency of these estimators can be investigated via convergence of solutions to the optimization problems. Assertions on the asymptotic distribution may be derived studying the ‘limit problem’ to appropriately ‘blown up’ problems.

Of course stability theory of stochastic programming can also be employed to prove results on so-called value-consistency of statistical models. Value consistency is considered, for instance, in [1].

From the beginning, in papers which use approaches from stochastic programming, emphasis has been put on epi-convergence instead of uniform convergence of the objective functions, which is often imposed in papers on asymptotic statistics ([2], [12], [13], [5], [8]). Constraints have also been considered, see [13], [4], [17] for statements about the asymptotic distribution in the case of constrained estimation.

Furthermore, papers on asymptotic statistics usually assumed that certain identifiability conditions are fulfilled, i.e. it is required that the optimization problems provide unique estimators. Meanwhile, however, also non-identifiable models gain growing interest from the statisticians’ side. Van der Vaart ([18]) presents a consistency result for M -estimators using Wald’s approach. Ferger [3] considers convergence in distribution and proves argmax theorems for the non-unique case. We will show by way of example how the rather general qualitative stability results for stochastic optimization problems in [19] and [21] can contribute to these investigations.

Our starting point is a brief discussion of (qualitative) stability assertions in the deterministic setting. Consistency results, i.e. the ‘in probability’ setting, will be described in more detail. The explanation of corresponding convergence notions ‘in distribution’ requires special preliminaries and cannot be given here because of the restricted number of pages. Application of the results in [21] to asymptotic statistics is considered in [22]. Strong consistency is studied in [9] and illustrated by examples.

The paper is organized as follows. In section 2 we quote van der Vaart’s consistency result for non-identifiable M -estimators ([18], Theorem 5.14). In section 3 we sketch stability theory in the deterministic setting and introduce the general random optimization models we shall deal with. Section 4 considers the ‘in probability’ version of the general stability results and shows how van der Vaart’s result can be derived and extended using stability theory of stochastic programming. Details which have to be omitted here are given in [22].

Measurability considerations will not be discussed in this paper. We will sloppily assume that all measurability conditions needed are satisfied.

2 Consistency of M -Estimators

The class of M -estimators is an important and widely investigated class of estimators which includes for instance maximum-likelihood estimators. If so-called identifiability conditions are not satisfied or hard to verify, the following consistency statement ([18]) may be used:

Let Θ and Z be metric spaces and let $(Z_i)_{i \in N}$ be a sequence of i.i.d. random variables, defined on a common probability space $[\Omega, \Sigma, P]$ with values in Z , provided with the σ -field of Borel sets $\sigma(Z)$. \tilde{P} denotes the probability measure which is induced on $\sigma(Z)$ by Z_i . For a function $m|\Theta \times Z \rightarrow \bar{R}$ and $n \in N$ define

$$M_n(\vartheta) := \frac{1}{n} \sum_{i=1}^n m(\vartheta, Z_i), \hat{\vartheta}_n := \arg \max_{\vartheta \in \Theta} M_n(\vartheta), \text{ and } M_0(\vartheta) := \int m(\vartheta, z) d\tilde{P}(z).$$

Uniqueness of $\hat{\vartheta}_n$, $n \in \{0, 1, \dots\}$, is not imposed. It is only assumed that $\Theta_0 := \{\vartheta_0 \in \Theta \mid \sup_{\vartheta \in \Theta} \int m(\vartheta, z) d\tilde{P}(z) = \int m(\vartheta_0, z) d\tilde{P}(z)\}$

is non-empty.

Theorem 1. (*Theorem 5.14 in [18]*)

Let the following assumptions be satisfied:

(A1) $m(\cdot, z)$ is upper semicontinuous in each $\vartheta \in \Theta$ for almost all z ,

(A2) for every sufficiently small ball $U \subset \Theta$ the function

$\sup_{\vartheta \in U} m(\vartheta, \cdot)$ is measurable and

$$\int \sup_{\vartheta \in U} m(\vartheta, z) d\tilde{P}(z) < \infty,$$

(A3) $M_n(\hat{\vartheta}_n) \geq M_n(\vartheta_0) - o_P(1)$ for some $\vartheta_0 \in \Theta_0$.

Then for each $\varepsilon > 0$ and for every compact set $K \subset \Theta$,

$$\lim_{n \rightarrow \infty} P(d(\hat{\vartheta}_n, \Theta_0) \geq \varepsilon \wedge \hat{\vartheta}_n \in K) = 0.$$

3 Stability of Stochastic Programming Problems

The general approach will be outlined in the deterministic setting. After that the definitions for convergence notions ‘in probability’ will be explained. We restrict ourselves to random variables Z_n with values in R^p . The results, however, hold also for random variables with values in a complete separable metric space.

We consider an optimization problem

$$(P_{0,D}) \quad \min_{x \in \Gamma_0} f_0(x)$$

which is approximated by surrogate problems

$$(P_{n,D}) \quad \min_{x \in \Gamma_n} f_n(x).$$

$\{f_n, n \in N \cup \{0\}\}$ is a family of objective functions $f_n| R^p \rightarrow \bar{R}^1$ and $\{\Gamma_n, n \in N \cup \{0\}\}$ is the corresponding family of closed constraint sets.

When dealing with constraint sets and solution sets which are not single-valued, convergence notions for sequences of sets are needed. Kuratowski-Painlevé-convergence has turned out to be the appropriate concept. For the behavior of the solution sets, however, usually one does not obtain full Kuratowski-Painlevé-convergence. Under reasonable conditions one can only expect, that cluster points of the solution sets to the approximate problems are solutions of the limit problem. This can be described in the following framework:

Let $(S_n)_{n \in N}$ be a sequence of subsets of R^p . Then the Kuratowski-Painlevé-Limes superior and the Kuratowski-Painlevé-Limes inferior are defined in the following way:

$$K\text{-}\limsup_{n \rightarrow \infty} S_n := \{s \in R^p | \exists (s_n)_{n \in N} : s_n \rightarrow s \text{ and } s_n \in S_n \text{ infinitely often}\},$$

$$K\text{-}\liminf_{n \rightarrow \infty} S_n := \{s \in R^p | \exists (s_n)_{n \in N} : s_n \rightarrow s \text{ and } s_n \in S_n \quad \forall n \geq n_0\}.$$

$$\text{If } K\text{-}\limsup_{n \rightarrow \infty} S_n = K\text{-}\liminf_{n \rightarrow \infty} S_n = S_0,$$

then the Kuratowski-Painlevé-Limes exists: $K\text{-}\lim_{n \rightarrow \infty} S_n = S_0$. For this statement we will use the abbreviation $S_n \xrightarrow{K} S_0$.

If $K\text{-}\limsup_{n \rightarrow \infty} S_n \subset S_0$, we will call the sequence $(S_n)_{n \in N}$ an inner approximation to S_0 and use the abbreviation $S_n \xrightarrow{K-i} S_0$. If $K\text{-}\liminf_{n \rightarrow \infty} S_n \supset S_0$, then $(S_n)_{n \in N}$ is an outer approximation to S_0 which will be abbreviated by $S_n \xrightarrow{K-o} S_0$.

A sequence $(S_n)_{n \in N}$ is e.g. an inner approximation to S_0 , if $K\text{-}\limsup_{n \rightarrow \infty} S_n$ is empty. Hence, when using assertions on inner approximations for the investigation of statistical estimates one has to impose additional conditions to ensure that there is a convergent (sub)sequence of estimates.

For single-valued $S_n, n \in N \cup \{0\}$, inner approximations reduce to convergence, if there is a compact set K such that $S_n \subset K \quad \forall n \geq n_0$. This compactness condition is to ensure that there exists a cluster point of $(S_n)_{n \in N}$. For outer approximations a corresponding result is valid without further conditions. Hence assertions on semiconvergence include assertions on convergence in the single-valued case.

The weakest condition on the objective functions (in the absence of constraints) which guarantees inner approximations is epi-convergence. In order to be able to deal also with constraints, often modified objective functions (which take the value $+\infty$ if the constraints are violated) are considered. Therefore we shall explain epi-convergence for objective functions which map into \bar{R}^1 .

The sequence $(f_n)_{n \in N}$ is said to be epi-convergent to f_0 if the sequence of the corresponding epigraph multifunctions converges in the Kuratowski-Painlevé sense to the epigraph of f_0 . The following condition is an equivalent

characterization of epi-convergence:

$\forall x_0 \in R^p :$

$$\begin{aligned} & (E1) \quad \forall (x_n)_{n \in N} \text{ with } x_n \rightarrow x_0 : \quad \liminf_{n \rightarrow \infty} \tilde{f}_n(x_n) \geq \tilde{f}_0(x_0), \\ \wedge \quad & (E2) \quad \exists (x_n)_{n \in N} \text{ with } x_n \rightarrow x_0 : \quad \limsup_{n \rightarrow \infty} \tilde{f}_n(x_n) \leq \tilde{f}_0(x_0). \end{aligned}$$

It should be mentioned that epi-convergence of the original objective functions together with Kuratowski-Painlevé-convergence of the constraint sets does not imply epi-convergence of the modified objective functions. Therefore results on simultaneous approximation of objective functions and constraint sets usually have to impose stronger convergence conditions on the original objective functions, in general continuous convergence.

One can 'divide' epi-convergence in the 'upper part' (E2), called epi-upper approximation, and the (more restrictive) 'lower' part (E1), called lower semicontinuous approximation. The denomination 'lower semicontinuous' approximation was chosen, because this is also the 'lower part' of continuous convergence. All types of convergence under consideration can be restricted to a 'convergence region' X , which is supposed to be closed. Such a restriction comes into play, because convergence is needed on the constraint set of the limit problem only. We will for instance speak of a lower semicontinuous approximation at X (abbreviated $f_n \xrightarrow{X} f_0$) if

$$\forall x_0 \in X \quad \forall (x_n)_{n \in N} \text{ with } x_n \rightarrow x_0 : \quad \liminf_{n \rightarrow \infty} f_n(x_n) \geq f_0(x_0).$$

A sequence of functions $(f_n)_{n \in N}$ which is a lower semicontinuous approximation to f_0 at X and for which $(-f_n)_{n \in N}$ is a lower semicontinuous approximation to $-f_0$ at X is continuously convergent to f_0 at X (abbreviated $f_n \xrightarrow{c, X} f_0$).

A pointwise convergent sequence of functions is an epi-upper approximation. Verification of lower semicontinuous approximations, however, often requires considerable effort. Hence sufficient conditions for this type of convergence are of special interest, see for instance [11].

The following theorem is a condensed version of stability results for the deterministic setting, see e.g. [19] for references and proofs in the almost surely setting.

Let Γ_n denote the constraint set, Φ_n the optimal value and Ψ_n the solution set to the problem $(P_{n,D})$, $n \in N \cup \{0\}$.

Theorem 2. (i) If $f_n \xrightarrow{c, \Gamma_0} f_0$, $\Gamma_n \xrightarrow{K} \Gamma_0$, and
 $\exists K \in C^p \quad \exists n_0 \in N \quad \forall n \geq n_0 : \quad \Gamma_n \subset K$,
 then $\Phi_n \rightarrow \Phi_0$.
 (ii) If $f_n \xrightarrow{\Gamma_0} f_0$ and $\Gamma_n \xrightarrow{K} \Gamma_0$, then $\Psi_n \xrightarrow{K-i} \Psi_0$.

Here C^p denotes the family of nonempty compact subsets of R^p . Of course corresponding results hold for the 'almost surely' case. The 'in probability' setting was studied in [19]. Semi-convergence in distribution of

optimal values and solution sets is the topic of [21]. Roughly spoken, stability results corresponding to the above theorem are available in these convergence modes. It has to be clarified, however, how appropriate convergence notions ‘in probability’ and ‘in distribution’ may be described. We will confine here to the ‘in probability’ setting.

The above theorem is not the weakest form. One could also take into account ε_n -optimal solutions and weaken the convergence condition in assertion (ii), see [22] for convergence in probability and in distribution and [9] for the ‘almost surely’ case.

4 Convergence in Probability

Now we turn to random optimization problems. In consistency investigations one usually has to deal with a deterministic limit problem. Stability results, however, are available also for random limit problems. Therefore we shall give the definitions for the general case. The advantage of having a deterministic limit problem can be exploited when looking for sufficient conditions for the convergence notions.

In the following we will assume that a random optimization problem

$$(P_0) \quad \min_{x \in I_0(\omega)} f_0(x, \omega)$$

is approximated by a sequence of random surrogate problems

$$(P_n) \quad \min_{x \in I_n(\omega)} f_n(x, \omega).$$

Here $[\Omega, \Sigma, P]$ is supposed to be a complete probability space, $I_n|\Omega \rightarrow 2^{R^p}$ is a closed-valued measurable multifunction (which will also be called random closed set) and $f_n|R^p \times \Omega \rightarrow R^1$ is a random function which is measurable with respect to $\mathcal{B}^p \otimes \Sigma$ (\mathcal{B}^p denotes the σ -field of Borel sets of R^p).

M -estimators fit into this framework with

$$f_0(x, \omega) = f_{0,D}(x) = - \int m(x, z) d\tilde{P}(z) \text{ and } f_n(x, \omega) = - \frac{1}{n} \sum_{i=1}^n m(x, Z_i(\omega)).$$

In the following we shall give the main definitions needed to prove stability results ‘in probability’. Details can be found in [19] and [10]. The definitions are based on investigations by Salinetti and Wets [15].

Let $\{G_n, n \in N_0\}$ be a family of random sets. $(G_n)_{n \in N}$ is called an inner approximation in probability to G_0 (abbreviated $G_n \xrightarrow{K-i-prob} G_0$) if $\forall \varepsilon > 0 \forall K \in \mathcal{C}^p : \lim_{n \rightarrow \infty} P([G_n \setminus U_\varepsilon G_0] \cap K \neq \emptyset) = 0$.

$(G_n)_{n \in N}$ is called an outer approximation in probability to G_0 (abbreviated $G_n \xrightarrow{K-o-prob} G_0$) if $\forall \varepsilon > 0 \forall K \in \mathcal{C}^p : \lim_{n \rightarrow \infty} P([G_0 \setminus U_\varepsilon G_n] \cap K \neq \emptyset) = 0$.

$(G_n)_{n \in N}$ is Kuratowski-Painlevé-convergent in probability to G_0 (abbreviated $G_n \xrightarrow{K-prob} G_0$) if it is an inner and an outer approximation.

For a family $\{f_n, n \in N_0\}$ of random functions and a closed subset X of R^p we say that $(f_n)_{n \in N}$ is a lower semicontinuous approximation in probability to f_0 at X (abbreviated $f_n \xrightarrow[X]{l-prob} f_0$) if $\forall \varepsilon > 0 \forall K \in C^{p+1}$:

$$\lim_{n \rightarrow \infty, l \rightarrow \infty} P\{\omega : (\text{Epi} f_n(\cdot, \omega) \cap [U_{\frac{1}{l}} X \times R]) \setminus U_\varepsilon(\text{Epi} f_0(\cdot, \omega) \cap [X \times R]) \cap K \neq \emptyset\} = 0.$$

Here $U_{\frac{1}{l}} X$ denotes the set $\{y \in R^p \mid \inf_{x \in X} \|x - y\| < \frac{1}{l}\}$.

An upper semicontinuous approximation can then be defined claiming that $(-f_n)_{n \in N}$ is a lower semicontinuous approximation for $-f_0$.

A stability theorem ‘in probability’ can be proved ([19]) which has the form of the above quoted stability theorem for the deterministic case with all convergence assumptions understood ‘in probability’ and a compactness condition of the form $\exists K \in C^p \forall \tilde{K} \in C^p : \lim_{n \rightarrow \infty} P(\Gamma_n \cap \tilde{K} \subset K) = 1$.

It is, however, to be mentioned that continuity of $f_0(\cdot, \omega)$ has to be imposed. This additional assumption results from general relations between convergence almost surely and convergence in probability in our framework.

If one deals with unconstrained problems, continuity and continuous convergence can be replaced with lower semicontinuity and epi-convergence.

ε_n -optimal solutions are not considered in [19]. The quoted stability statements can, however, also be applied to this case, making use of the following lemma, which is proved in [22].

Lemma 1. *Let $(\varepsilon_n)_{n \in N}$, $\varepsilon_n \mid R^p \times \Omega \rightarrow R^1_+$, be a sequence of random functions such that*

$$\sup_{x \in U_{\kappa} \Gamma_0} \varepsilon_n(x, \cdot) \xrightarrow{prob} 0 \text{ for a suitable } \kappa > 0.$$

Then for $(f_n)_{n \in N}$ and a sequence $(g_n)_{n \in N}$ with $g_n := f_n + \varepsilon_n$ we have $f_n \xrightarrow[\Gamma_0]{l-prob} f_0 \Rightarrow g_n \xrightarrow[\Gamma_0]{l-prob} f_0$.

Thus, with a suitably chosen sequence $(\varepsilon_n)_{n \in N}$, approximate estimators can be regarded as true optimizers of g_n , and the stability results can be applied also to the case of approximate estimators.

There are sufficient conditions for the convergence assumptions, which open the possibility to reduce convergence of sequences of random functions to convergence of suitable sequences of random variables. The approach was called pointwise approach in [20], [11] and scalarization in [6]. It is applicable also to dependent sequences of random variables. If a deterministic probability measure is approximated by the empiric measure as in section 2, the assumptions (A1) and (A2) imply epi-convergence in probability ([22]).

References

1. Dudley, R. M. (1998) Consistency of M-estimators and one-sided bracketing. In: High Dimensional Probability (E. Eberlein, M. Hahn, M. Talagrand, eds.), Birkhäuser, Basel, 151–189

2. Dupačová, J., Wets, R. J.-B. (1988) Asymptotic behavior of statistical estimators and of optimal solutions of stochastic problems. *Ann. Statist.* 16, 1517–1549
3. Ferger, D. (2003) A continuous mapping theorem for the argmax-functional in the non-unique case. *Statistica Neerlandica* 58, 1–14
4. Geyer, C. J. (1994) On the asymptotics of constrained M-estimation. *Ann. Statist.* 22, 1993–2010
5. Hess, C. (1996) Epi-convergence of normal integrands and strong consistency of the maximum likelihood estimator. *Ann. Statist.* 24, 1298–1315
6. Korf, L.A., Wets, R. J.-B. (2001) Random lsc functions: an ergodic theorem. *Math. Oper. Res.* 26, 421–445
7. King, A. J., Rockafellar, R. T. (1993) Asymptotic theory for solutions in statistical estimation and stochastic programming. *Math. Oper. Res.* 18, 148–162
8. Knight, K. (2003) Epi-convergence in distribution and stochastic equi-semicontinuity. Technical Report, University of Toronto
9. Lachout, P., Liebscher, E., Vogel, S. (2003) Strong convergence of estimators as ε_n -estimators of optimization problems. Preprint M 01/03, TU Ilmenau
10. Lachout, P., Vogel, S. (2003) On continuous convergence and epi-convergence of random functions. Part I: Theory and relations. *Kybernetika* 39, 75–98
11. Lachout, P., Vogel, S. (2003) On continuous convergence and epi-convergence of random functions. Part II: Sufficient conditions and applications. *Kybernetika* 39, 99–118
12. Pflug, G. C. (1992) Asymptotic dominance and confidence for solutions of stochastic programs. *Czechoslovak J. Oper. Res.* 1, 21–30
13. Pflug, G. C. (1995) Asymptotic stochastic programs. *Math. Oper. Res.* 20, 769–789
14. Prékopa, A. (1994) *Stochastic Programming*. Kluwer Academic Publishers
15. Salinetti, G., Wets, R. J.-B. (1986) On the convergence in distribution of measurable multifunctions (random sets), normal integrands, stochastic processes and stochastic infima. *Math. Oper. Res.* 11, 385–419
16. Shapiro, A. (1989) Asymptotic properties of statistical estimators in stochastic programming. *Ann. Statist.* 17, 841–858
17. Shapiro, A. (2000) On the asymptotics of constrained local M-estimators. *Ann. Statist.* 28, 948–960
18. van der Vaart, A. W. (1998) *Asymptotic Statistics*. Cambridge University Press
19. Vogel, S. (1994) A stochastic approach to stability in stochastic programming. *J. Comput. and Appl. Math., Series Appl. Analysis and Stochastics* 56, 65–96
20. Vogel, S. (1995) On stability in stochastic programming - Sufficient conditions for continuous convergence and epi-convergence. Preprint TU Ilmenau
21. Vogel, S. (2003) On semicontinuous approximations of random closed sets with application to random optimization problems. *Ann. Oper. Res.* (to appear)
22. Vogel, S. (2004) What can asymptotic statistics gain by stability theory of stochastic programming? Preprint M 01/04, TU Ilmenau
23. Wets, R. J.-B. (1989) Stochastic programming. In: *Handbooks in Operations Research and Management Science*, Vol. 1, Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North Holland, 573–629.

Time Lags in Capital Accumulation

Ralph Winkler^{1,3}, Ulrich Brandt-Pollmann², Ulf Moslener¹, and Johannes P. Schlöder²

¹ Interdisciplinary Institute for Environmental Economics,
Bergheimer Str. 20, 69115 Heidelberg, Germany

² Interdisciplinary Center for Scientific Computing,
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

³ Corresponding author:

phone: ++49-6221-548019, fax: ++49-6221-548020, email: winkler@uni-hd.de

Abstract. We formulate an optimal control capital accumulation model with an exogenously given time lag between investment and the accumulation of the capital stock to analyze the qualitative and quantitative influence of time lags to the system dynamics. Optimal investment paths for a finite time lag are shown to be cyclic as opposed to the monotone paths for instantaneous capital accumulation. Furthermore, we show how to reformulate the retarded differential equation in a suitable way so that sophisticated numerical methods for optimal control can be applied to analyze the impact of the length of the time lag. It turns out that both the frequency and the amplitude of the cycles depend on the length of the investment period.

1 Introduction

We know from Austrian capital theory [5]: all production takes time. That is, the transformation of inputs into outputs does not occur instantaneously. However, the common neoclassical theory usually abstracts from the time structure of the production process. Exceptions are [4] who derive a generalized maximum principle for exogenously given and constant time lags between control and state variables. Following an idea first posed in [6], [8] empirically analyze in how far time lags between investment and capital accumulation can explain business cycles. [1] show that the system dynamics of time lagged capital accumulation models can exhibit limit-cycles.

Different from the authors mentioned above, we explicitly analyze the *qualitative* and *quantitative* properties of the optimal paths. Therefore, we formulate an optimal control capital accumulation model with an exogenously given time lag between investment and the accumulation of capital in section 2. Hence, the dynamics of our capital accumulation model is governed by a system of *functional differential equations* (section 3). In section 3.1 optimal investment paths for a finite investment period are shown to be cyclic as opposed to the monotone paths for instantaneous capital accumulation. Furthermore, we present a systematic analysis of the impact of the length of the investment period by numerical investigations. We show how to reformulate

the retarded differential equation in a suitable way so that sophisticated numerical methods for optimal control can be applied in section 3.2. In section 4 optimal paths of capital and investment for different lengths of the investment period for the resulting high-dimensional optimal control problem are calculated. Section 5 concludes.

2 The Model

We analyze an optimal control capital accumulation model similar to the neoclassical growth models with one commodity in [3] and [7]. Suppose the following intertemporal welfare function W to be maximized

$$W\{c(t)\} = \int_0^\tau V(c(t)) \exp[-\rho t] dt ,$$

where ρ denotes the positive and constant discount rate, τ the finite or infinite time horizon and V the twice differentiable, monotonic increasing ($V' > 0$) and concave ($V'' < 0$) instantaneous welfare function. The commodity is produced according to a twice differentiable production function P which is a monotonic increasing ($P' > 0$) and non concave ($P'' \leq 0$) function of the capital stock k . The produced amount of commodity can either be used for consumption c or investment i in the capital stock k .

To model the time structure of production we assume that capital accumulation is time consuming: investment at time t is supposed to accumulate the capital stock delayed at time $t+\sigma$, where σ denotes the positive and constant time lag between investment and capital accumulation. Furthermore, we assume the capital stock deteriorates at the positive and constant rate γ .

Assuming further that the capital stock k cannot be consumed, i. e. $i(t) \geq 0$, the optimal control problem reads

$$\begin{aligned} \max_{i(t)} \int_0^\tau V(c(t)) \exp[-\rho t] dt \quad & \text{s.t.} \\ c(t) &= P(k(t)) - i(t) , \\ \dot{k}(t) &= i(t-\sigma) - \gamma k(t) , \\ i(t) &\geq 0 , \\ i(t) &= \xi(t) , \quad t \in [-\sigma, 0[, \\ k(0) &= k_0 . \end{aligned} \tag{1}$$

Because of the retarded differential-difference equation, the specification of an initial value for the capital stock k is not sufficient for a unique solution. In addition, we have to specify an initial path ξ for the investment i in the time interval $[-\sigma, 0[$.

3 Optimal Investment Path

To determine the necessary conditions for an optimal solution the generalized maximum principle in [4] is applied. Hence, the present value Hamiltonian \mathcal{H} reads

$$\mathcal{H} = V(c(t)) \exp[-\rho t] + p_c(t) [P(k(t)) - i(t) - c(t)] + p_k(t+\sigma)i(t) - p_k(t)\gamma k(t) + p_i(t)i(t) ,$$

where p_c and p_i denote the Kuhn-Tucker parameters of the corresponding restrictions and p_k the costate variable of the capital stock k . Assuming that \mathcal{H} is continuous differentiable with respect to i , the necessary conditions for an optimal solution read

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial i(t)} &= -p_c(t) + p_k(t+\sigma) + p_i(t) = 0 , \\ \frac{\partial \mathcal{H}}{\partial c(t)} &= V_c(c(t)) \exp[-\rho t] - p_c(t) = 0 , \\ \frac{\partial \mathcal{H}}{\partial k(t)} &= p_c(t)P_k(k(t)) - p_k(t)\gamma = -\dot{p}_k(t) , \\ p_i(t) &\geq 0 , \quad p_i(t)i(t) = 0 . \end{aligned}$$

As the Hamiltonian \mathcal{H} is concave in k and i these necessary conditions are also sufficient, if in addition, the following transversality condition is satisfied

$$\lim_{t \rightarrow \tau} [p_k(t)k(t)] = 0 . \quad (2)$$

3.1 Analytical Analysis of the Optimal Paths

Assuming an interior solution, i. e. $p_i(t) = 0$ for all t , the following system of functional differential equations for an optimal solution can be derived

$$\begin{aligned} \dot{c}(t) &= \frac{V_c(c(t))}{V_{cc}(c(t))}(\gamma + \rho) - \frac{V_c(c(t+\sigma))}{V_{cc}(c(t))} (P_k(k(t+\sigma)) \exp[-\rho\sigma]) , \\ \dot{k}(t) &= P(k(t-\sigma)) - c(t-\sigma) - \gamma k(t) . \end{aligned} \quad (3)$$

Although this system is not analytically solvable in general, we can state some qualitative properties of its solution. First, we determine the fix point (c^*, k^*) of system (3) by setting $\dot{c} = \dot{k} = 0$. Proposition 1 states the result.

Proposition 1 (Stationary State). *If the production function P is strictly monotonic increasing ($P' > 0$) and satisfies the Inada conditions ($\lim_{k \rightarrow 0} P' = \infty$, $\lim_{k \rightarrow \infty} P' = 0$) then the system of functional differential equations (3) has a unique fix point (c^*, k^*) , which is unambiguously determined by the following equations:*

$$\begin{aligned} c^* &= P(k^*) - \gamma k^* , \\ P'(k^*) &= (\gamma + \rho) \exp[\rho\sigma] . \end{aligned}$$

The economic interpretation is straight forward. In the stationary state consumption c equals the stationary state production minus the replacement of deteriorated capital. Furthermore, the marginal productivity of capital P' in the stationary state has to be the higher the longer the time lag σ to compensate for the waiting period until new investment turns out to be productive. As a consequence, if marginal productivity is strictly decreasing ($P' < 0$) then the stationary state capital stock is the smaller the longer the time lag σ .

Second, we examine the system dynamics in a neighborhood around the stationary state. Therefore, we linearize the system of differential equations (3) around (c^*, k^*) derived from proposition 1. Introducing the abbreviations $A = \gamma + \rho$, $B = \frac{V'(c^*)}{V''(c^*)} P''(k^*) \exp[-\rho\sigma]$ and $C = (\gamma + \rho) \exp[\rho\sigma]$ the characteristic equation reads

$$0 = x^2 + x\{\gamma + A(\exp[\sigma x] - 1) - C \exp[-\sigma x]\} + A\gamma \exp[\sigma x] + AC \exp[-\sigma x] - A\{\gamma + C\} - B. \quad (4)$$

For $\sigma = 0$ the characteristic equation (4) reduces to a quadratic equation with two real solutions. For $\sigma > 0$ the equation has in addition an infinite number of complex solutions. It can be shown that proposition 2 holds [12].

Proposition 2 (Roots of the characteristic polynomial). *For positive constants A , B and C the characteristic equation (4) has a unique negative real solution $x_1 < 0$ and at least one positive real solution $x_2 > 0$. If the time lag is positive $\sigma > 0$, equation (4) has in addition an infinite number of complex solutions with negative real part and an infinite number of solutions with positive real part.*

As there are solutions with positive and solutions with negative real part the system dynamics of the (linearized) system of differential equations exhibits saddle point stability, as the space of solutions decomposes in a stable manifold spanned by the eigenvectors corresponding to the eigenvalue with negative real part and an unstable manifold spanned by the eigenvectors corresponding to the eigenvalues with positive real part. Similar to the case of ordinary first-order differential equations the optimal paths for consumption c and the capital stock k in the linear approximation around the stationary state are given by $c(t) = c^* + \sum_n c_n \exp[x_n t]$ and $k(t) = k^* + \sum_n k_n \exp[x_n t]$, where x_n denote the characteristic roots, i.e. the solutions of equation (4), as well as the c_n and k_n denote constants which can (at least in principle) be unambiguously determined by the set of initial conditions k_0 , ξ and the transversality condition (2).

As a consequence, for an infinite time horizon τ the optimal solution converges asymptotically towards the stationary state to satisfy the transversality condition (2). According to the turnpike theorem the system converges also for finite time horizons towards the stationary state at the beginning, stays in a neighborhood of the stationary state for most of the time and diverges from the stationary state at the end of the time horizon if the time

horizon is sufficiently long [11]. Furthermore, the system dynamics exhibits a qualitative change by a transition from $\sigma = 0$ to $\sigma > 0$. In the first case the optimal paths converge strictly monotonic and exponentially, in the latter case cyclical and exponentially damped towards the stationary state.

3.2 Numerical Analysis of the Optimal Paths

In order to numerically solve (1), we reformulate the problem to make it accessible to the advanced optimal control package MUSCOD-II¹ [9] which exploits the multiple shooting state discretization [2], [10]. First, we split the fixed time interval of length τ into n parts each of the length of the time lag σ and obtain the optimal control problem

$$\max_i \int_0^\sigma \sum_{l=1}^n V(c_l(t)) \exp[-\rho(t + \sigma(l-1))] dt \quad \text{s.t.}$$

$$\begin{aligned} \dot{k}_1(t) &= f(t, k_1(t), i_0(t), p), \\ &\vdots \\ \dot{k}_n(t) &= f(t, k_n(t), i_{n-1}(t), p), \end{aligned}$$

with the path constraints

$$\begin{aligned} c_l(t) &= P(k_l(t)) - i_l(t), \\ i_l(t) &\geq 0, \quad l = 1, 2, \dots, n. \end{aligned}$$

The history is given by $i_0(t) = \xi(t + \sigma)$ ($t \in [-\sigma, 0]$). To guarantee the continuity of the differential variables after each time interval σ we need to introduce linearly coupled multipoint constraints

$$r^s(s_l^k, p) + r^e(s_{l-1}^k, p) = 0, \quad l = 2, 3, \dots, n.$$

In this context $r^s(k_l, p)$ gives the initial conditions for k_l and $r^e(k_{l-1}, p)$ the end point conditions for k_{l-1} , respectively. Hence, we obtain a boundary value problem.

Second, each time interval of length σ is split into m multiple shooting subintervals on which the controls are chosen to be constant or piecewise linear which makes the optimal control problem to become a finite dimensional one. The resulting optimal control problem is of the dimension $(n-1) \times m = M$ which is solved using a direct multiple shooting method implemented in the package MUSCOD-II.

¹ MUSCOD-II was developed by the Simulation and Optimization group at the Interdisciplinary Center for Scientific Computing, University of Heidelberg.

4 An Example: Linear Limitational Production Function

In the following, we show numerical optimizations for a linear limitational production function. Indeed, a linear production function does not satisfy the requirements of proposition 1. Therefore, in general a unique, positive and finite stationary state does not exist. To ensure a unique, positive and finite stationary state we assume further that production needs another non-producible input factor besides capital, e. g. labor, which is given in constant amount \bar{l} . Hence, production is bounded from above by the amount of available labor \bar{l} and, as long as the accumulation of capital turns out to be optimal, we achieve a unique, positive and finite stationary state. The reason why we choose a linear limitational production function is twofold. First, if P is linear then $B = 0$ and the characteristic equation simplifies, which allows at least a graphical determination of the characteristic roots as demonstrated in [12]. Second, the stationary state does not depend on the time lag σ . As a consequence, the numerical optimizations we show for different time lags σ are easier to compare with each other.

In general, a linear production function has the form $P(k(t)) = \alpha_0 + \alpha_1 k(t)$ ($\alpha_0 \geq 0$, $\alpha_1 > 0$). A positive constant α_0 indicates that the commodity can also be produced by the non-producible factor alone without using capital. Without loss of generality, we assume that one unit of labor alone produces one unit of the commodity and λ units of labor are used together with κ units of capital to produce one unit of the commodity, indicated by $\alpha_0 = \bar{l}$ and $\alpha_1 = (1 - \lambda)/\kappa$. Hence, the stationary state is given by $c^* = \bar{l}/(\lambda + \kappa\gamma)$ and $k^* = \kappa\bar{l}/(\lambda + \kappa\gamma)$. Note that in the stationary state the whole amount of labor is used to employ and maintain the capital stock.

Figure 1 shows the results of the numerical optimization, given a linear limitational production function.² The time lag σ varies from 0 to 2.5. As supposed the optimal paths converge monotonic towards the stationary state for instantaneous capital accumulation ($\sigma = 0$). Most of the time both paths remain in a direct neighborhood around the stationary state. Because of the finite time horizon investment turns to zero near the end of the time horizon. As a consequence, also the capital stock drops. The larger the time lag σ the more the optimal paths differ from the optimal paths achieved by instantaneous capital accumulation and the more cyclical patterns can be observed. The optimal paths converge cyclical towards the stationary state at the beginning and diverge cyclical from the stationary state at the end of the time horizon. The question remaining is, why are cyclical paths optimal, i. e. better than monotone paths if $\sigma > 0$. Due to the cyclical "swing-in" towards and "swing-out" from the stationary state the economy can achieve a longer phase where all labor is solely used to employ capital which is more

² The following functions and constants have been chosen: $V(c(t)) = \ln c(t)$, $\bar{l} = 26\frac{2}{3}$, $\lambda = 0.8$, $\kappa = 0.3$, $\gamma = 0.15$, $\rho = 0.1$, $\tau = 25$, $k_0 = 0$ and $\xi(t) = 0$.

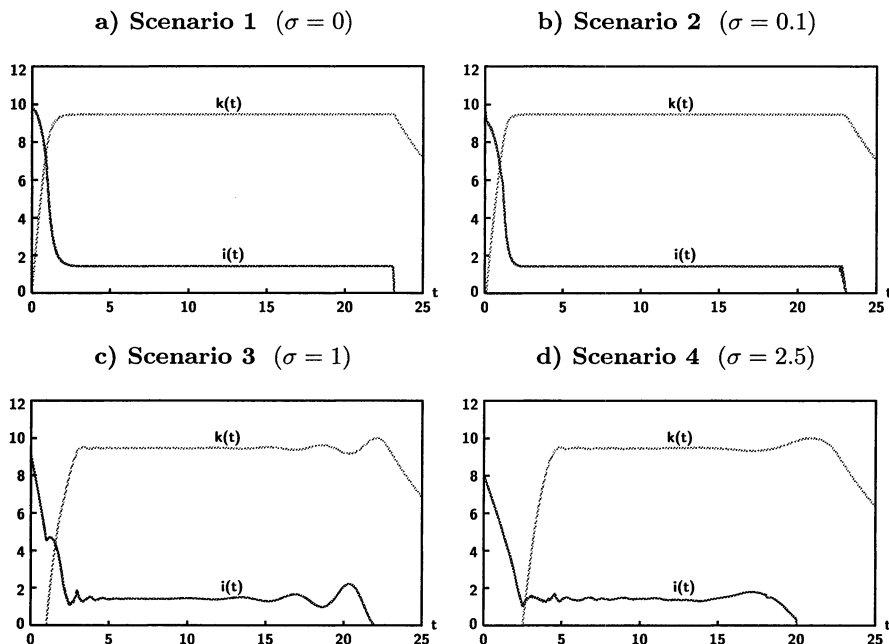


Fig. 1. Optimal consumption and capital paths for different time lags between investment and capital accumulation. The numerical optimization has been executed by the MUSCOD-II program package. In the present application we obtain up to 2500 degrees of freedom! The computing time on a 2533 MHZ Intel P4 machine running under Linux varies between 2400 and some 9000 seconds strongly depending on the actual number of controls (about 96 % of the computing time is consumed for solution of condensed QPs).

efficient than producing the commodity by labor alone. Beside the inertia caused by the time lag there is another source of inertia, the constancy of the capital stock. As negative investment is ruled out, disinvestment can only take place by depreciation and is therefore more tedious than investment. As a consequence the frequency for the "swing-out" is smaller than the frequency for the "swing-in".

5 Conclusion

We have shown that time lagged capital accumulation models exhibit in general a *qualitatively* different system dynamics compared to instantaneous capital accumulation models. While the optimal paths of the latter converge strictly monotonic towards the stationary state, the first show cyclical and exponentially damped optimal paths. Furthermore, we have shown how to *quantitatively* analyze the system dynamics with sophisticated numerical tools like MUSCOD-II. Our calculation results suggest that the optimal paths change

continuously from strictly monotone to cyclical and exponentially damped convergence for an increasing time lag σ .

As a consequence, the standard assumption of instantaneous capital accumulation in neoclassical economic theory can be justified as good approximation for small time lags σ . On the contrary, for large time lags, e.g. in the plant construction or pharmaceutical industry, the validity of this approximation is endangered.

So far we have analyzed a centralized economy. A priori it is not clear if the well known result of the non retarded accumulation model that a decentralized market solution is Pareto-optimal turns out to be true in the case of time lagged capital accumulation. Intuition suggests that the households' knowledge about the time lag might be crucial. To answer this question further investigations on this topic have to be carried out.

References

1. Asea, P.K., Zak, P.J. (1999) Time-to-build and cycles. *Journal of Economic Dynamics and Control* 23, 1155–1175
2. Bock, H.G., Plitt, K.J. (1984) A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings of the 9th IFAC World Congress*, Budapest. Pergamon Press
3. Cass, D. (1965) Optimum growth in an aggregative model of capital accumulation. *Review of Economic Studies* 32, 233–240
4. El-Hodiri, M.A., Loehman, E., Whinston, A. (1972) An optimal growth model with time lags. *Econometrica* 40, 1137–1146
5. Faber, M., Proops, J., Speck, S. (1999) Capital and Time in Ecological Economics – Neo-Austrian Modelling. Edward Elgar, Cheltenham Northampton
6. Kalecki, M. (1935) A macroeconomic theory of business cycles. *Econometrica* 3, 327–344
7. Koopmans, T.C. (1965) On the concept of optimal economic growth. In: *Study week on the econometric approach to development planning*. North Holland, Amsterdam, 225–266
8. Kydland, F.E., Prescott, E.C. (1982) Time to build and aggregate fluctuations. *Econometrica* 50, 1345–1370
9. Leineweber, D.B. (1999) Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models. *Fortschritt-Berichte VDI, Reihe 3, No. 613*. VDI Verlag GmbH, Düsseldorf
10. Leineweber, D.B., Bauer, I., Bock, H.G., Schlöder, J.P. (2003) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization – part I: theoretical aspects. *Comput. Chem. Engng* 27, 157–166
11. McKenzie, L.W. (1981) Optimal economic growth, turnpike theorems and comparative dynamics. In: Arrow, K.J., Intriligator, M.D. (Eds.): *Handbook of Mathematical Economics*. North-Holland, Amsterdam New York Oxford Tokyo, 1281–1355
12. Winkler, R. (2003) Zeitverzögerte Dynamik von Kapital- und Schadstoffbeständen – Eine österreichische Perspektive. Metropolis, Marburg

First-Price Bidding and Entry Behavior in a Sequential Procurement Auction Model

J. Philipp Reiß and Jens Robert Schöndube

Faculty of Economics and Management, University of Magdeburg, Germany

Abstract. We introduce a procurement auction model where capacity-constrained firms face a sequence of two procurement auctions, each of them in the first-price sealed-bid design. Our main findings are that the firms' entry decisions depend on relative project completion cost levels and that equilibrium bidding in both auction stages deviates from the standard Symmetric Independent Private Value auction model (SIPV) due to opportunity costs of bidding created by possibly employed capacity. The model highlights the fact that firms with identical completion costs for the first project may differ in entry and bidding strategies. In addition, experimental data is reported in order to assess the predictive power of the model.

1 Introduction

Theoretical contributions to the analysis of procurement auctions tend to focus on the allocation of single projects ignoring outside options of firms, e.g. [14], [10], [4], [2], [3]. However, recent empirical research on repeated procurement auctions suggests that outside options in the sense of additional future procurement auctions of similar projects affect firms' entry and bidding behavior in real life, see [5] and [7]. E.g., the study [7] finds that a firm which did not win a highway procurement contract earlier in a sequence of auctions run by the Californian Department of Transportation (1994-1998) is twice as likely to enter a subsequent auction than a firm which already won a (large) contract. This evidence suggests that firms are aware of their opportunity costs of bidding created by employed capacity. Thus, firms seem to be choosy with respect to entry if facing an auction sequence of non-identical procurement contracts and might include these opportunity costs in their submitted bids.

This paper studies this kind of entry decision in the context of privately known completion cost rankings and analyzes how firms refine their bidding strategies with opportunity costs of early bid submission. Our main theoretical findings are that the entry decision depends on relative project completion cost levels and equilibrium bidding in both auction stages deviates from the standard SIPV model and its sequential formulation with homogenous goods due to opportunity costs of bidding created by possibly employed capacity. It follows from the analysis that firms with identical completion costs for the first project may differ in entry and bidding strategies.

In addition, this paper presents experimental evidence for our sequential procurement auction model that suggests that its theoretical implications

might explain observed patterns of entry behavior in repeated real-life procurement auctions. In turn, this finding points to an explicit role for the auction environment in empirical studies on static procurement auctions.

2 Model and Equilibrium Behavior

We consider two risk-neutral firms with capacity to complete a single project due to capacity constraints. Completion costs for two subsequently auctioned projects, L and M , are private information to each firm. It is common knowledge that firm i 's costs of completion are jointly drawn from $f(l_i, m_i)$ with domain $[\underline{c}, \bar{c}]$ and stochastic equivalent in the sense of $f(l, m) = f(m, l)$ for every $(l, m) \in [\underline{c}, \bar{c}]^2$ implying $E[L_i] = E[M_i]$. It follows that each firm faces either lower completion cost for project L or project M . If cost realizations of a firm are such that $l < m$, this firm is said to have a cost advantage for project L , the reversed inequality indicates a cost advantage for project M . Although completion costs of a single firm may be correlated across projects, pairs of completion costs of different firms are independently distributed.

In each procurement auction, a participating firm may submit a sealed bid where the lowest bid wins the project and the amount bid is paid in exchange for completion of the project. However, bids cannot exceed maximum completion costs \bar{c} which may be interpreted as the procurer's outside option, otherwise infinite prices result. We assume that the auctioneer cannot set a price below maximum completion costs \bar{c} and that resale of projects is not feasible. If there happens to be a bidding tie, any auctioneer employs a fair chance mechanism to break it. The sequence of auctions begins with the procurement auction of project L where the winner is announced before project M is auctioned off. Thus with two firms, any firm can infer if it faces competition in auction M before it submits its bid.

In addition to equilibrium bidding functions for each auction stage, b^L and b^M , a firm's strategy also includes a decision to submit a bid in the first auction or skip bidding for project L . In general, firm 1 participates in auction L if its expected profit from bidding exceeds the opportunity cost arising from possibly being excluded from bidding for project M , formally

$$E[\Pi_1^{L+M} | (l_1, m_1)] \geq E[\Pi_1^M | (l_1, m_1)],$$

where $E[\Pi_1^{L+M} | (l_1, m_1)]$ denotes firm 1's expected profit if it bids in the procurement auction for project L and - if unsuccessful - continues bidding in auction M and $E[\Pi_1^M | (l_1, m_1)]$ is its expected profit if it skips the first auction and bids only for the subsequently auctioned project M .

The firm's decision to submit a bid in auction L depends on the relation of its project completion costs. In order to formalize the entry decision, we introduce the critical-value function $g_1: [\underline{c}, \bar{c}] \rightarrow [\underline{c}, \bar{c}]$ which specifies a cut-off point for completion costs of project L , depending on completion costs of project M , beyond which it is not worthwhile for firm 1 to participate in the

auction of project L . Specifically, the decision rule to submit a bid for project L is given by entry function $\varepsilon(\cdot)$ as follows:

$$\varepsilon_1(l_1, m_1) = \begin{cases} \text{Enter Auction } L & \text{if } l_1 \leq g_1(m_1) \\ \text{Skip Auction } L & \text{if } l_1 > g_1(m_1), \end{cases}$$

where the critical-value function $l_1^{crit} = g_1(m_1)$ is implicitly defined by the equality of expected profit from entering auction L and corresponding opportunity cost:

$$E[\Pi_1^{L+M} | (l_1^{crit}, m_1)] = E[\Pi_1^M | (l_1^{crit}, m_1)]. \quad (1)$$

By definition, firm 1 is indifferent between entering auction L and skipping it if its completion cost pair satisfies $l_1^{crit} = g_1(m_1)$. In symmetric equilibrium, equation (1) is given by (for details see [12]):

$$\begin{aligned} 0 = & \frac{\bar{c} - m}{2} + \frac{2\bar{c} - [g(m) + m]}{2} \left[\frac{1}{2} - \int_{\underline{c}}^{\bar{c}} \int_{g(y)}^{\bar{c}} f(x, y) dx dy \right] \\ & + [\bar{c} - g(m)] \int_{\underline{c}}^{\bar{c}} \int_{g(y)}^{\bar{c}} f(x, y) dx dy - (\bar{c} - m) \int_{\underline{c}}^{\bar{c}} \int_{\underline{c}}^{g(y)} f(x, y) dx dy \\ & - \int_m^{\bar{c}} \int_{g(y)}^{\bar{c}} (y - m) \cdot f(x, y) dx dy. \end{aligned}$$

The properties of the critical-value function $g(m)$ in symmetric equilibria are summarized in the next proposition that is proved in [12].

Proposition 1. *For all symmetric perfect Bayesian equilibria characterized by the representative firm's strategy $[b^L(l, m), b^M(m), \varepsilon(m)]$ and the density function $f(l, m)$:*

- (a) *There exists a (nonempty) compact and convex set of completion cost pairs where it is rational for a firm to bid for project L although it has a cost advantage for completing project M . This subset is defined by $G = \{(l, m) \in [\underline{c}, \bar{c}] | m \leq l \leq g(m)\}$.*
- (b) *The critical value function $g(m)$ exists and*
 - (i) $m < g(m) < \bar{c}$ f. $m \in [\underline{c}, \bar{c})$, $g(\bar{c}) = \bar{c}$, $g(\underline{c}) > \underline{c}$,
 - (ii) $g'(m) > 0$,
 - (iii) $g''(m) < 0$ f. $m \in [\underline{c}, \bar{c})$ and $g''(\bar{c}) = 0$.

It follows from proposition 1 that a firm always enters auction L if it faces a cost advantage for this project, i.e. $l \leq m$. Then it can earn at least $\bar{c} - m$. In

contrast, a cost advantage for completing project M implies the impossibility of the firm to secure itself the same return in auction L as it could earn in auction M being the only bidder. However, if the competing firm skipped auction L , too, then there is competition in the second auction with the risk of low or even zero profits due to aggressive bidding. Therefore, a firm with lower cost for project M may wish to participate in the first auction and win the unloved project L at a high price to insure itself against low profits resulting from fierce competition in the second auction, although it actually prefers losing the auction for project L . Figure 1 illustrates these results for a representative firm with completion costs (l, m) .

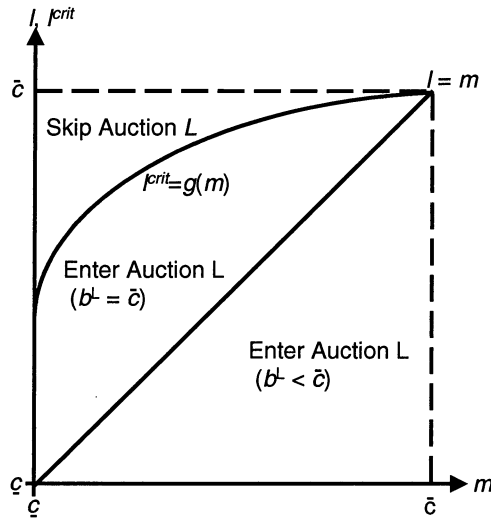


Fig. 1. A Typical Critical-Value Function $g(m)$:

If $l \leq g(m)$, the firm submits a bid in the first auction. A firm with relatively high cost for completing the first project, i.e. $l > g(m)$, does not submit a bid.

The symmetric equilibrium bidding strategy in a one-shot first-price auction is well-known. For a discussion see e.g. [9] and [16]. However, in our dynamic setting additional issues arise. Consider first the auction for project L . Any of the two firms participating in the first auction anticipates that if it does not win project L , it remains the only bidder in the subsequent auction M and receives $\bar{c} - m$. Thus, it may submit a very large bid for project L since it is, at least partially, insured against losing auction L . A firm with a cost advantage for project M knows that it cannot achieve such a high return in auction L as it would receive by winning the second auction after it lost the first one. Thus, provided the firm decides to participate in auction L , it minimizes chances of winning project L by submitting the highest feasible bid. In contrast, if a firm has a cost advantage for project L , it tops its com-

pletion cost l with its certain return from auction M and uses the revised cost parameter $\tilde{l} = l + \bar{c} - m$ in auction L .

If there is no bidding competition in the auction for project M , then any firm bidding for it will submit the maximum feasible bid to maximize profits. In case of bidding competition, the updated belief concerning the competitor's cost parameter for project M takes into account that it also skipped auction L which may not be equilibrium behavior for every type. The appropriate a posteriori pdf is denoted by $f_{M|Skip}(m)$ and gives the (equilibrium) density that a firm with completion cost realization m for project M bids only in auction M .

The equilibrium bidding functions are summarized in the next proposition, for a proof see [12].

Proposition 2. *Equilibrium Bidding Functions in Auctions L and M*
The equilibrium bidding functions of a firm with cost pair $(l, m) \in [\underline{c}, \bar{c}]^2$ are given by:

$$b^L(\tilde{l}) = \begin{cases} \bar{c} & \text{if } l \geq m \\ \tilde{l} + \frac{\int_{\tilde{l}}^{\bar{c}} [1 - F_L(x)] dx}{1 - F_L(\tilde{l})} & \text{otherwise} \end{cases}$$

if it submits a bid for project L where $F_L(x) = \int_{\underline{c}}^x \int_{\bar{c} + \underline{c} - \tilde{l}}^{\bar{c}} f(m - \bar{c} + \tilde{l}, m) dm d\tilde{l}$ with $x, \tilde{l} \in [\underline{c}, \bar{c}]$ and $\tilde{l} = l + \bar{c} - m$, and

$$b^M(m) = \begin{cases} \bar{c} & \text{if it is the only bidder} \\ m + \frac{\int_m^{\bar{c}} [1 - F_{M|Skip}(x)] dx}{1 - F_{M|Skip}(m)} & \text{otherwise} \end{cases}$$

if it submits a bid for project M where $g(x)$ denotes the competitor's critical-value function, $f_{M|Skip}(x) = \int_{g(x)}^{\bar{c}} f(y, x) dy / \int_{\underline{c}}^{\bar{c}} \int_{g(s)}^{\bar{c}} f(y, s) dy ds$ and

$$F_{M|Skip}(x) = \int_{\underline{c}}^x f_{M|Skip}(s) ds.$$

3 Impact of Auction Environment on Bidding

In this section, we briefly demonstrate how bidding behavior in the first auction varies with the value that a firm places on the opportunity to participate in a second auction. This option value itself depends on a firm's completion cost for the second project. For the purpose of illustration, consider a specific example with two firms where completion costs are distributed uniformly: $(l, m) \sim U[20, 100]^2$.

- (a) (Base Scenario): Suppose the completion cost for project M to be fixed at some level below maximum completion cost, say $m = 80$. The bidding function for the first project L is

$$b^L(l, m = 80) = \begin{cases} \frac{2}{3} \cdot \frac{l^3 + 30l^2 - 1,664,000}{l^2 - 12,800} & l < 80 \\ 100 & l \geq 80. \end{cases}$$

Figure 2 illustrates that bids in the dynamic auction environment with $m = 80$ substantially exceed bids in a one-shot auction determined by $b(l)$ given below.

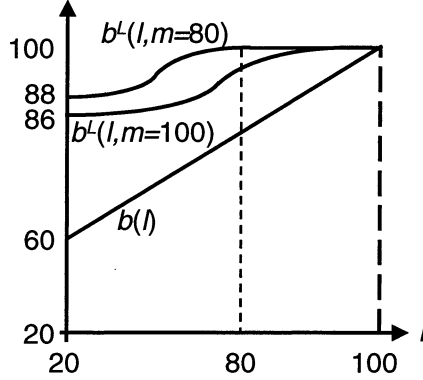


Fig. 2. Bidding for Project L under Various Outside Options

- (b) (One-Shot Auction) There is no second auction such that the first auction reduces to the standard SIPV bidding function in a procurement context:

$$b(l) = 50 + 0.5l$$

- (c) (Virtually No 2nd Auction): Consider base scenario (a) with maximum completion cost $m = 100$. Compared to $b^L(l, m = 80)$, the larger completion cost m shifts the bidding function $b^L(l, m = 100)$ downwards,

$$b^L(l, m = 100) = \frac{2}{3} \cdot \frac{l^3 - 30l^2 - 1,660,000}{l^2 - 12,400 - 40l}.$$

The exemplified change of $b^L(l, m)$ in response to increases in m illustrates typical comparative static behavior. Note that bids with virtually no second auction do not coincide with bids in the one-shot auction since completion costs are private information.

4 Experimental Results

The preceding section highlighted the fact that variations of the second round auction game change the equilibrium bidding function in the first round. In particular, larger cost for completing the second project, which correspond to a lower value of the second round game, lead to more aggressive bidding for the first project. This theoretical prediction and that on entry behavior are fulfilled in the laboratory implementation of our auction model. We sketch some of the results that are obtained by Brosig and Reiß in a study

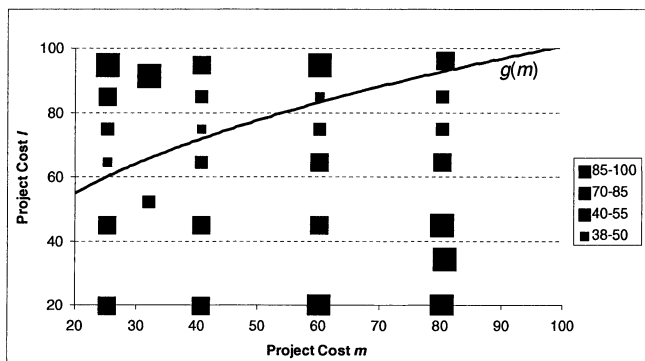


Fig. 3. Entry Decisions in Line with Theory

of sequential procurement auctions at the Magdeburg Laboratory for Experimental Economics (MaxLab), see [1].

Figure 3 summarizes data on entry generated in one of the treatments where 24 subjects played the procurement auction sequence, each of them with 28 pairs of completion costs. Each marking in the (m, l) -space represents one completion cost pair. The size of the markings indicates the frequency of decisions on entry in line with the theoretical prediction relative to all observed decisions on entry for that cost pair. Out of a total of 672 observed entry decisions, approximately 70% are correctly predicted by our theory. It

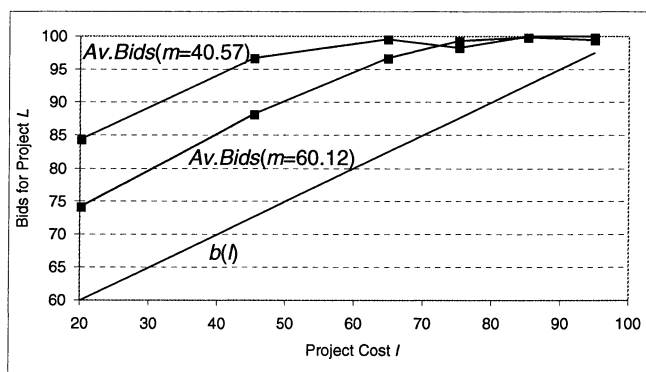


Fig. 4. Observed Bidding for Project L

is apparent from figure 3 that the number of correctly predicted entries rises as the cost pair is farther away from the critical-value function. In the vicinity of $g(m)$, a false entry decision is less costly and on its graph, the cost is zero by definition. Thus, the theory's predictive power increases as the expected cost of incorrect entry is larger, which is rigorously confirmed in [1].

Figure 4 depicts average bids for project L as a function of l for two different cost levels for the second project. Apparently, observed bids for $m=40.57$ (significantly) exceed those for $m=60.12$ in the relevant range 20–60.12 which is in line with our sequential procurement auction model, see section 3. Both bid samples (significantly) exceed predictions of the standard one-shot auction model. Since the overbidding phenomenon in standard auction experiments translates to underbidding in the procurement auction context, this fact is particularly remarkable and strengthens the suggestion that subjects include opportunity costs of early bidding in their bids for project L .

References

1. Brosig, J., Reiß, J. P. (2003): Entry Decisions and Bidding Behavior in Sequential First-Price Procurement Auctions: An Experimental Study. FEMM Working Paper No. 03023, University of Magdeburg
2. Budde, J., Göx, R. F. (1999): The Impact of Capacity Costs on Bidding Strategies in Procurement Auctions. *Review of Accounting Studies* 4, 5–13
3. Celentani, M., Ganuza, J. J. (2002): Corruption, and competition in procurement. *European Economic Review* 46, 1273–1303
4. Dasgupta, S., Spulber, D. F. (1988): Managing Procurement Auctions. *Information Economics and Policy* 4, 5–29
5. De Silva, D., Dunne, T. et al. (2002): Sequential bidding in auctions of construction contracts. *Economics Letters* 76, 239–244
6. Gale, I. L., Hausch, D. B. (1994): Bottom-Fishing and Declining Prices in Sequential Auctions. *Games and Economic Behavior* 7, 318–331
7. Jofre-Bonet, M., Pesendorfer, M. (2000): Bidding Behavior in a Repeated Procurement Auction. *European Economic Review* 44, 1006–1020
8. Kagel, J. H. (1995): Auctions: A Survey of Experimental Research. In: Kagel, J. H., Roth, A. E. (Eds.): *Handbook of Experimental Research*, Princeton, NJ: Princeton University Press, 501–580
9. McAfee, R. P., McMillan, J. (1987a): Auctions and Bidding. *Journal of Economic Literature* 25, 699–738
10. McAfee, R. P., McMillan, J. (1987b): A reformulation of the principal-agent model. *Rand Journal of Economics* 18, 296–307
11. Milgrom, P. R., Weber, R. J. (1982): A Theory of Auctions and Competitive Bidding. *Econometrica* 50, 1089–1122
12. Reiß, J. P., Schöndube, J. R. (2002): On Participation in Sequential Procurement Auctions. FEMM Working Paper No. 02016, University of Magdeburg
13. Riley, J. G., Samuelson, W. F. (1981): Optimal Auctions. *American Economic Review* 71, 381–392
14. Riordan, M. H., Sappington, D. (1987): Awarding Monopoly Franchises. *American Economic Review* 77, 375–387
15. Weber, R. J. (1983): Multiple Object Auctions. In: Engelbrecht-Wiggans, R., Shubik, M., Stark, R. (Eds.): *Auctions, Bidding and Contracting: Uses and Theory*, New York, NY: New York University Press, 165–191
16. Wolfstetter, E. (1995): Auctions: An Introduction. *Journal of Economics Surveys* 10, 367–420

Financial Market Web Mining with the Software Agent PISA

Patrick Bartels and Michael H. Breitner

Institut für Wirtschaftsinformatik, Universität Hannover

Königsworther Platz 1, D-30167 Hannover, Germany

Email: bartels@iwi.uni-hannover.de and breitner@iwi.uni-hannover.de

Abstract. The World Wide Web (WWW) contains millions of hypertext pages that present nearly all kinds of information. Specific effort is required to use this information as input to subsequent computations or to aggregate values in web pages. To do so, data from webpages are extracted and stored on computers either in text files or databases. These manipulations can be better attained by using an autonomous software program instead of human personal. Here, we present a platform independent software agent called PISA (Partially Intelligent Software Agent) that extracts financial data autonomously from webpages and stores them on a local computer. Data quality has highest significance. PISA generates time series with user defined denseness. These time series have adequate quality for financial market analyses, e. g. forecasts with neural networks.

1 Introduction

Financial market websites contain financial market information from banks, brokers, exchanges and financial service providers, e. g. stock quotes, option prices, exchange rates in particular and other information concerning exchange markets. The internet offers these data in many cases near time. Comparable up-to-dateness usually is only offered by commercial finance databases that must be paid. Fees are usually high. The internet often offers the same information cost free. Here, up-to-dateness and cost-freeness are exploited to build a financial database for free. The database's quality is as good as of commercial financial databases. Financial market websites usually do not contain static but dynamic content. The presented data change frequently. To minimize work the presented data are usually stored in databases. When a webpage is requested by a browser the needed information is queried from the database and the results are put into a specific template. Templates are a skeleton of the resulting pages which contain specific tags. These tags define where specified pieces of information are filled in. This results in highly structured webpages. Since usually one single template is used for many kinds of webpages, once a scheme is recognized, this scheme can be used to identify the demanded information on other pages of the same website. Here, schemes

are used to extract specified information to generate not just a database but dense time series. A time series is defined as a series or function of a variable over time. This means that a particular variable takes a particular discrete value at a sequence of points in time. Here, quotes are used as variables. Time series can be used to train artificial neural networks. E. g. the FAUN¹ neurosimulator project uses neural networks to predict real market option prices, see [2], and to make short term forecasts, see [5]. Those neural networks need input data with highest data quality.

Here, we present the platform independent software agent PISA to generate time series. The resulting time series are capable of being used for neural network training. Major problems of web-mining that usually effect output data quality are prevented. Primary problem is that most webpage's structure changes frequently and the position of information might change. In this case either no information or wrong information might be extracted. No information is as well extracted if a website is not available for a certain time. In cases information is distributed to different webpages, data from different webpages must be aggregated to realize added value, e. g. a price difference for arbitrage trades. To enable time series with flexible denseness the agent has to extract wanted information in adequate small intervals. As the Internet is an international medium websites are formatted with several international formats like for date, time and numbers. Most existing web-mining programs are designed to extract the text representing these data. To assure highest data quality date, time and numbers need to be consistently formatted. The agent's data output limits further processing possibilities. Therefore, both text files and databases have to be supported to store the extracted data.

2 Software-Agents

2.1 Agent Paradigm

Primary advantage of using an agent is that agents can work very efficiently 24 hours a day and 7 days a week (except breakdowns and maintenance work). Agents can handle dozens of extraction operations per minute automatically. Able to work long periods of time with low costs software agents are well qualified for web-mining tasks.

There is no generally accepted definition of the term *software agent* although the difference between normal programs and software agents has been discussed intensely for many years now [cf. 4, 3]. A common characterization was disposed by Jennings and Woodridge [cf. 7]. Their definition is the origin for almost all current research on agent technology. Accordantly an agent is a representative who works on behalf of a person and has the following attributes: *Autonomy* – The agent should be able to execute the assigned tasks on its own without any call-

¹ FAUN = Fast Approximation with Universal Neural networks

back. *Social behavior* – Agents interact with other agents and at least with the user. *Ability to react* – Perception of the system environment the agent is "living" in and the ability to react on basis of more or less precisely defined decision patterns. *Consciousness* – An agent does not only react on events but can anticipate future incidents. The proficiency of the four characteristics depends on the agent's aims. Here, the agent has to receive and process webpages automatically. Autonomy is required to work over a long period of time without necessary user interaction. User interaction is however necessary for configuration. This requires only little social behavior. The agent has to react on the perceived situation. Consciousness is not necessary since all necessary decisions can be made using hard coded rules. Case differentiations within the program code are sufficient because all possible cases are a priori known. As the structure of the processed webpages is known in advance the agent just has to filter user specified patterns without "thinking". This does not suffice to call the agent intelligent. The presented agent is called partially intelligent and its intelligence will be further developed.

2.2 Agent Requirements

Usually financial market webpages contain several pieces of independent information on a single website, e. g. stock quotes of a specific index. These information chunks have to be correctly and reliably identified, extracted and saved. Each step requires specific abilities of a web-mining agent.

Receiving webpages: To receive a webpage a request is send to a webserver using TCP/IP-protocols which have to be supported. If the information containing webpage's URL (Uniform Resource Locator) is not a priori known crawling methods are mandatory. The agent must recognize if a webpage contains relevant information or not. Here, for neural network training a continuous data flow is mandatory. The agent has to be permanently available during trading hours. Referring to the need of efficiency the agent should only work if it is reasonable. Financial websites are only updated when a change of the underlying data occurs. In times with only few changes processing frequency can be decreased. Timer functions enable reasonable system utilization. Extraction intervals must be as small as possible to enable time series with optimal denseness. Minimal system utilization has high priority.

Extraction: Dealing with HTML (Hypertext Markup Language) documents the structure is not always completely defined and can be irregular. HTML documents can contain errors. Missing tags are a common example for these errors. Since browser programs handle these problems they are often not noticed by users and/or webmasters. The agent has to recognize and handle such problems to assure error tolerant HTML parsing. Once a webpage's source code is received and parsed regular patterns are advisable to identify and extract complex patterns from HTML source code. String Tokenizer methods are less powerful than regular expressions but they are usually faster. Both should be supported.

The internet offers financial data from all over the world. Dependent on the target group different international number, time and date formats are used. These

have to be recognized and reformatted in user defined formats. This increases flexibility for further processing programs. Some important information, e. g. date and time of a quote, is often split into multiple pieces. The agent has to be able to merge them. Therefore, as well as for adjusting different time zones date, time and number information has to be transformed into numbers. All extracted values have to be accessible as variables to calculate new values and to merge distributed items.

Data storage: File handling methods are mandatory to save extracted patterns. The user should be able to choose an output file format. Both plain text files and XML-files (XML = Extensible Markup Language) should be supported. Large amounts of data can be handled easier stored in a database instead of text files. The most common protocol for accessing databases is the ODBC-Protocol (Open database connectivity) which should be supported by the programming language as well.

Beside the mentioned requirements there are some general ones. The agent should be as platform independent as possible to assure flexible application. The hazard of breakdowns has to be minimized to assure a continuous data flow. Input data has to be correct. Accurateness is a very important aim. The agent has to provide rules with which extracted data can be checked on plausibility.

2.3 Existing Agents

Web-mining agents are developed for several years now. Usually web-mining agents and web-crawlers are developed for a specific task. This explains several drawbacks which are summarized here.

All considered programs are able to request and receive webpages. Missing timer functionality prevents that extraction tasks can be executed in specified intervals. Performance is further affected by missing multithreading support. Most agents support adequate crawling methods. Usually regular patterns are used to identify and extract pieces of information. The relative position of a pattern to another pattern is usually not supported. The position within the source code can only be defined by generating a complex pattern. The HTML structure is hardly taken into account. The resulting extraction rules are vulnerable for errors. Even if the right value is correctly extracted often no formatting methods are provided. Also most wrapper tools are only capable of saving the extracted results in plain text files. Only few of them support XML. ODBC-support is very seldom.

The mentioned drawbacks do not appear for all considered agents. No program is fully capable for the given task. Upgrading existing programs fails since the considered programs are usually not well documented and the source code is not accessible. Inadequate extensibility and missing functions are sufficient reasons to develop an own web-mining agent. An overview for public domain and commercial web-mining agents is accessible in [6].

3 Software Agent PISA

3.1 Design and Implementation

The mentioned requirements lead to special demands for the language PISA is programmed in. The major considered languages and their attributes are shown in Table 1 that summarizes the results of a detailed consideration [cf. 1]. The languages are differently applicable. Most missing functions can be extended using public domain modules. Since these programs are usually not well documented the language should support the features inherently. Here, Java is the most suitable programming language to achieve the mentioned goals.

Table 1. Overview of considered programming languages.

↕ Functions Languages ⇨	PHP	Java-Script	Perl	C++	C#	Java
Internet functions	✓	✓	✓	✓	✓	✓
Regular expressions	✓	✓	✓	✓	✓	✓
String tokenizer functions	✓	✓	✓	✓	✓	✓
Multithreading	✗	✗	✓	✓	✓	✓
Error handling	✗	✗	✗	✓	✓	✓
Timer functions	✗	✗	✗	(✗)	✓	✓
File handling	✓	✗	✓	✓	✓	✓
ODBC database support	✓	✗	✓	✗	✓	✓
Remote method invocation	✗	✗	✗	✓	✓	✓
Platform independency	✓	✓	✓	✓	✗	✓

PISA is completely realized in Java and consists of five components, shown in Figure 1. The component *PisaMain* initiates and starts all user defined extraction tasks. Specifications are taken from a configuration file. The initiated tasks run as independent threads and are executed simultaneously. Each task is represented by a single *PisaCrawler* object. These objects request the defined webpages and approve that during the crawling process no webpage is requested multiple times. Each received webpage is process by the *HtmlDocument* component that parses the passed website's source code. The source code is further processed by the *PisaGrabber* component which identifies and returns the user wanted patterns. The extracted data are saved either in a plain text file, XML-file or a database.

3.2 Functionality

Due to the dynamic nature of the web, most information extraction systems focus on specific extraction tasks. Here, we concentrate on agent based generation of dense time series. The specific problems are focused here.

PisaMain: PISA starts with executing the main module *PisaMain* that initiates the extraction tasks. They are defined by the user in a file with a specific syntax. The *PisaMain*-object starts one *PisaCrawler*-object per URL. These *Pisa*-

Crawler-objects are started with a one second time delay, each. For dense time series the request interval is very small. If too many pages are requested from a webserver too frequently, the webserver might crash or it does not answer some requests. Latter results from a standard mechanism to avoid webserver overload by ignoring requests when a specified number of requests in a period is exceeded. This results in information gaps in the time series. The delay time is adjustable.

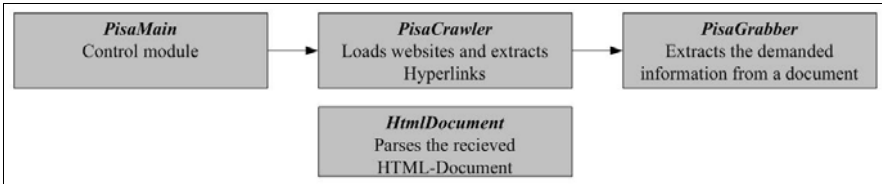


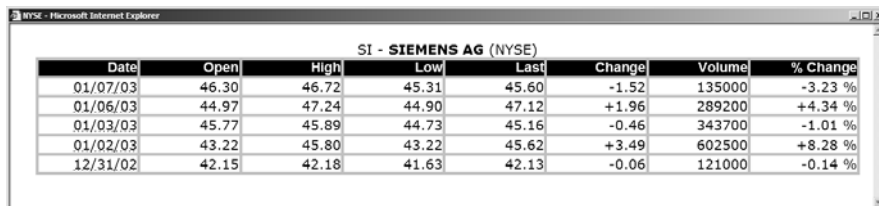
Fig. 1. Major modules of PISA.

PisaCrawler: Each PisaCrawler-object is executed in an adjustable interval. The interval is defined in seconds to enable dense time series. Shorter intervals are possible but not necessary since financial quotes are updated at most every second. Without given interval the PisaCrawler-object terminates itself after the first evaluation.

The PisaCrawler component crawls websites either at every execution or just once at the first run. In this case PISA recognizes interesting webpages by user defined requirements and memorizes the URL. Future accesses use this address. Each requested webpage is represented by an HtmlDocument-object and is further processed by the PisaGrabber-object.

HtmlDocument: The evoking class passes an URL. The HtmlDocument component requests and analyzes the according webpages. PISA handles common syntactical errors reliably. Therefore, an error tolerant HTML parsing process is realized. The source code is converted into XHTML compliant text. Afterwards, all needed tags are identified using regular expressions for start- and end-position of a tag. Only needed tags are processed to decrease processing time. Once a tag is identified its attributes like size, color and content are analyzed and stored in a tag-object. This tag-object represents the HTML tag. For each kind of tag an array is created that contains the objects of a kind in order of appearance. This enables a successive comparison of the array fields and the search for a special pattern.

PisaGrabber: The PisaGrabber module extracts the demanded patterns from a passed HtmlDocument-object. The easiest way to extract a pattern is successive comparison of each array field with the wanted pattern. This approach is neither comfortable nor capable because the number of appearance might change. If a stock's bid-price is appearance number two of a 2-digit number today, it can be the third one tomorrow. Here, the number of a requested pattern is defined relative to an anchor. This anchor is also defined by a pattern. An example clarifies this procedure: A typical table with stock prices is shown in Figure 2. The last given price can be found by using a pattern for a 2-digit decimal number. The current date can be used as the anchor. The current price's position is the forth table cell after the current date. Both patterns are specified by regular expressions.



Date	Open	High	Low	Last	Change	Volume	% Change
01/07/03	46.30	46.72	45.31	45.60	-1.52	135000	-3.23 %
01/06/03	44.97	47.24	44.90	47.12	+1.96	289200	+4.34 %
01/03/03	45.77	45.89	44.73	45.16	-0.46	343700	-1.01 %
01/02/03	43.22	45.80	43.22	45.62	+3.49	602500	+8.28 %
12/31/02	42.15	42.18	41.63	42.13	-0.06	121000	-0.14 %

Fig. 2. An example of a webpage presenting the quotes of the Siemens AG at the NYSE.

Using this approach the user has to know four things: 1. The pattern of the requested information; 2. The anchor pattern; 3. The number of tags between anchor-pattern and wanted information; 4. The kind of tag the patterns are formatted with. Optionally names for each extracted bit of information can be defined. This enables storage in platform independent XML-files. The names can be used like variables either to calculate new values or to merge several bits of information, e. g. a complete date that is split into data and time. If the user specifies a time zone PISA is working in, extracted worldwide times are converted in local time.

To assure high data quality, accurateness of the extracted data is very important. Information items can be defined as mandatory. If an object is declared as mandatory and not available on a webpage the whole dataset is abolished. In some cases quotes are published with a certain time delay. If the quote date is not available the current date is extracted. Time of quote and date of visit might not match. E. g. if the time delay is 10 minutes the current time is 08/17/03 12:05 am when the quote time is 11:55 pm the day before. Merging current date and time of quote results in a future time, i. e. 08/17/03 11:55 pm. PISA accomplishes this problem. Additionally rules check the extracted data if they fit certain criteria, e. g. it has to be within a specified fluctuation margin. Extracting data from several webpages leads to the problem that the display format of numbers and dates do most likely differ from each other. PISA formats text, numbers and dates in adjustable formats to facilitate import in retailing programs.

3.3 Examples

PISA was tested in detail in [1]. Here, we summarize the results. We tested the agent in different environments and for different tasks. The underlying website specific schemes have been generated manually.

To test the crawling function we decided not only to use financial websites because the URL of specific information is usually a priori known. Therefore eBay-customer profiles were generated. Starting from an eBay-user's feedback page the agent followed only the links to auction pages. These links were recognized by a specified pattern given by the user. PISA extracted auction details. Such user profiles are interesting for cross-selling activities.

To test PISA's performance we extracted prices for 60 German options from three different websites, see [1]. All options were extracted from different pages. The 180 pages were processed with an interval of 2 minutes each. The system

worked reliably and stable. Because of the resulting system utilization a high-capacity computer is recommended to keep processing times adequate. The resulting time series are used for generating real market option prices, see [2].

PISA extracted currency exchange rates for US Dollar and Euro from 06/03/03 to 07/16/03 from Finanztreff (www.finanztreff.de) with an interval of 10 seconds. Beside the current bid- and ask-price also date, time and German security identification number were reliably extracted. The results are used for short term forecasts. As the results in [5] show the data quality is very high. At most 2 or 3 values are missing in a row in the time series. These gaps are filled by using linear approximations. Reasons for missing values are most likely network failures.

4 Conclusion

Tests showed that PISA generates high quality data sets, e. g. for neural network training. Even a high number of webpages can be processed in short intervals. Intervals can be defined by the user approximately. The process is only limited by the computer's performance the agent is running on.

Even if the extraction process works well some limitations are left that affect data quality. Network failures, server sided blackouts and maintenance work make page requests impossible. Another source of errors are significant layout changes of a webpage. Today, only little changes can be anticipated by PISA.

5 References

1. Bartels P., Breitner M. (2003): Automatic Extraction of Derivative Market Prices from Webpages using a Software Agent. *IWI Discussion Paper Series No. 4*, Institut für Wirtschaftsinformatik, Universität Hannover, p. 1 - 32.
2. Breitner M. (1999): Heuristic Option Pricing with Neural Networks and the Neurocomputer Synapse 3, *Optimization* 81, 319 - 333.
3. Brenner W., Zarnekow R. and Wittig H. (1998): *Intelligent Software Agents: Foundations and Applications*. Springer, Berlin.
4. Caglayan A. (1997): *Agent Sourcebook: A Complete Guide to Desktop, Internet, and Intranet Agents*. John Wiley & Sons, Wien.
5. Mettenheim H.-J. (2003): Entwicklung der grob granularen Parallelisierung für den Neurosimulator FAUN 1.0 und Anwendungen in der Wechselkursprognose. Dissertation, Hannover.
6. Tredwell R., Kuhlins S. (2003): Wrapper Development Tools. <http://www.wifo.uni-mannheim.de/~kuhlins/wrappertools/>.
7. Woodridge M. and Jennings N. (1995): *Intelligent Agents: Theory and Practice*, *Knowledge Engineering Review* 10, 115 - 152.

Non-Linear Programming Solvers for Decision Analysis

Xiaosong Ding¹, Mats Danielson², and Love Ekenberg³

¹ Department of Information Technology and Media, Mid-Sweden University, 851 70, Sundsvall, Sweden. E-mail: cris.ding@mh.se

² Department of Informatics/ESI, Örebro University, 701 82, Örebro, Sweden. E-mail: msdn@esi.oru.se

³ Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology, Forum 100, Kista, 164 40, Stockholm, Sweden. E-mail: lovek@dsv.su.se

Abstract. Several methods have been developed over a number of years for solving decision problems when vague and numerically imprecise information prevails. However, the DELTA method and similar methods give rise to particular bilinear programming problems that are time consuming to solve in a real-time environment. This paper presents a set of benchmark tests for non-linear programming solvers for solving this special type of problems. With two existing linear programming based algorithms, it also investigates the performance of linear programming solvers for special decision situations in decision analysis systems.

1 Introduction

With the rapid development of graphical user interfaces, it is possible to bring the use of sophisticated computational techniques for decision analysis to a broader group of users, and many decision analytic tools have emerged. However, most of them consist of some straightforward set of rules applied to precise numerical estimates of probabilities and values no matter how unsure a decision maker (DM) is of his/her estimates. The requirement to provide numerically precise information in such models has often been considered unrealistic in real-life decision situations. Besides, sensitivity analysis is often not easy to carry out in more than a few dimensions at a time because of precise figures. When a DM is faced with a decision problem that could not be directly judged by his/her empirical experience, or according to available historical data, a module allowing imprecision is obviously of great value.

A number of techniques allowing imprecise statements have been suggested, but they focus more on representation and less on evaluation. In spite of several years of intense activities, only a few decision tools, for example, ARIADNE, *DecideIT* and Winpre, can evaluate imprecise estimates. Among these tools, the DELTA method, which was proposed in [2], acts as the basic theory for *DecideIT*, and has been used for decision problems like [3], [4], [6], [7], [8], by allowing the analysis of decision situations containing imprecise

information represented as intervals and relations. The purpose of this paper is to investigate the performance of modern combined linear programming (LP)/non-linear programming (NLP) solvers when we try to extend DELTA into the evaluation of general decision situations.

The next section describes some contents of DELTA related to this paper, analyzes the special bilinear programming (BLP) issue arising from DELTA, and investigates some related work. The penultimate section is devoted to the simulation study, which is followed by conclusions and future work.

2 The DELTA Method

The decision tree in Fig. 1 represents a decision situation where $D1$ is a decision node, and $E1$ and $E2$ are probability nodes representing indeterminism with associated probability distributions. The leaves are consequence nodes with associated value or utility functions.

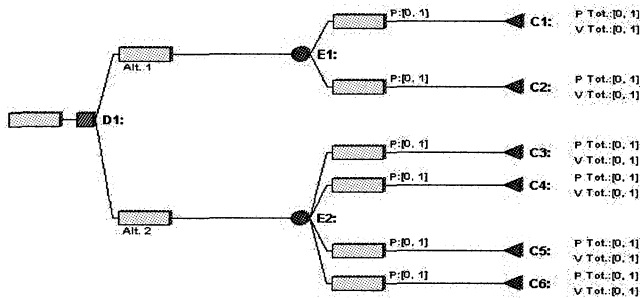


Fig. 1. A Decision Situation

In DELTA, a *decision frame* represents a decision problem. The idea behind such a frame is to collect all information necessary for the model in one structure. This structure is then filled in with user statements represented as linear inequalities. User statements can be a range constraint, $x_i \in [a, b]$, a comparative constraint, $x_i - x_j \in [a, b]$, a difference constraint, $(x_i - x_j) - (x_k - x_l) \in [a, b]$, and a compound constraint, $x_{h_1} + \dots + x_{h_m} \in [a, b]$ with respect to probability and value, respectively. All these statements are translated and collected together in a *probability base* (P -base) and a *value base* (V -base). The structure $\langle P, V \rangle$ is then referred to as a *decision frame*.

Given a *decision frame* $\langle P, V \rangle$, the primary evaluation rules in DELTA are based on pairwise comparisons using a generalization of the principle of maximizing the expected utility. Taking the decision situation as shown in Fig. 1, a typical problem is to maximize the expression:

$$\max \left(\sum_{k=1}^2 p_{1k} v_{1k} - \sum_{k=1}^4 p_{2k} v_{2k} \right)$$

$$s.t. \begin{bmatrix} L_P \\ L_V \end{bmatrix} \leq \begin{bmatrix} C_P & 0 \\ 0 & C_V \end{bmatrix} \cdot \begin{bmatrix} P \\ V \end{bmatrix} \leq \begin{bmatrix} U_P \\ U_V \end{bmatrix} \quad (1)$$

where P and V consist of variables coming from P -base and V -base; k represents the index of the consequences within an alternative; C_P and C_V are translated linear inequalities to express imprecise statements regarding P -base and V -base; and L and U are lower and upper bounds for those inequalities. (1) can be easily mapped into the standard quadratic programming problem.

To analyze (1), firstly, assuming that imprecise statements appear only in one base, while the other has precise estimates, (1) reduces to an LP system. Next, if imprecise information exists in both bases, (1) gives rise to a BLP problem because of its quadratic term $p \cdot v$. Moreover, when trying to compute the difference between the expected values of two alternatives, we could also obtain a negative value. This fact means that (1) is indefinite, and thus leads to the global optimization area. Since we prefer to take the two-phase strategy as in [11], this benchmark focuses on the local optimization phase.

Our first choice is the revised active set method because *DecideIT* needs to maintain a feasible solution during the calculation. Today, research on this issue chiefly concentrates on solving large-scale optimization problems rather than relatively small and sequential problems, according to the contraction method in DELTA, to handle sensitivity analysis. Hence, some earlier work seems more valuable. The original attempts to solve a BLP problem by applying the active set method dates back to [1], [5], [10], [12], and more. Recent work, for example, [9], focuses on how to update the matrices in use efficiently and make use of the previous information. This is the essential point that causes the revised active set method to be fast. Nevertheless, just as there are two sides to a coin, it is also this point that causes the revised active set method to be slow because there is only one constraint entering or leaving the active set at every step. Some other research work has already attacked (1) successfully. As proposed in [2], two LP based algorithms, Probability Bilinear Optimization (PBO) and Value Bilinear Optimization (VBO), can reduce (1) into an LP system, but a DM has to rule out certain statements.

3 Simulation Study

3.1 Preconditions

The NLP and LP solvers are SNOPT and SQOPT, respectively, which were originally developed by Systems Optimization Laboratory (SOL), Stanford University. All simulation study work takes place under Win2000, Matlab6.5, Tomlab3.1, PentiumIII 1000MHz processor and 512M memory.

3.2 Comparisons between PBO and VBO

PBO and VBO have very similar structures except for 2 normalization constraints in the P -base. We apply SNOPT and SQOPT to the data sets gen-

erated according to the demand of PBO and VBO, respectively. This experiment makes use of 10 groups of data for PBO. Each group consists of instances from 51 to 250 consequences where each instance has around $2N$ constraints, making a total of $10 \times 200 = 2000$ simulated decision situations. By analogy, there are also 2000 instances applied to VBO. Experimental results are demonstrated in Fig. 2.

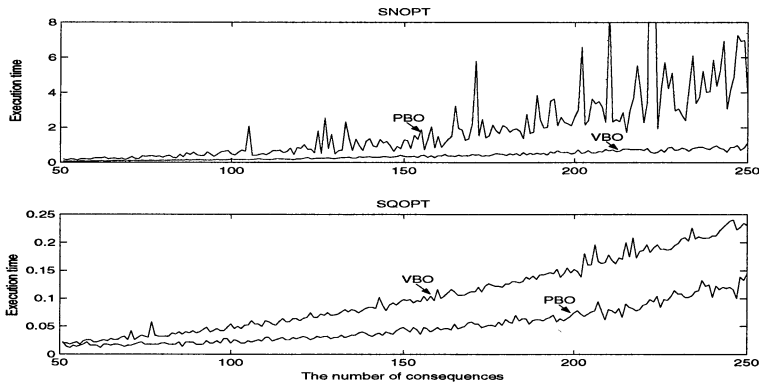


Fig. 2. PBO vs. VBO

It can be observed that for these two specially defined decision situations, SNOPT has a better and more stable performance for VBO than for PBO; while on the other hand, SQOPT indicates that to solve a PBO problem is faster. Both facts suggest there must be some difference between PBO and VBO. This difference could serve as one of the research directions. If we could realize and eliminate this difference, a generalized version of an LP based algorithm could be obtained.

3.3 Comparisons between BLP and LP

With PBO and VBO, it is possible to compare BLP with LP. The purpose is to identify how large the gap is between them. The same data sets are applied as in the last experiment. Figure 3 demonstrates their performance. Table 1 provides numerical results. It divides the data into 4 sections and calculates the average execution time of these 10 groups of data. The gap indicates how much faster the LP algorithm is compared to the BLP algorithm.

It can be observed that for VBO, the gap between LP and BLP algorithms is around 3.5. This is because SQOPT performs more poorly and SNOPT performs better for VBO as indicated in Fig. 2. Nevertheless, the situation reverses for PBO, which makes its gap much more than 10. Consequently, the research work for a better BLP algorithm seems reasonable. Moreover, as (1) contains such a special structure, this also makes the idea possible.

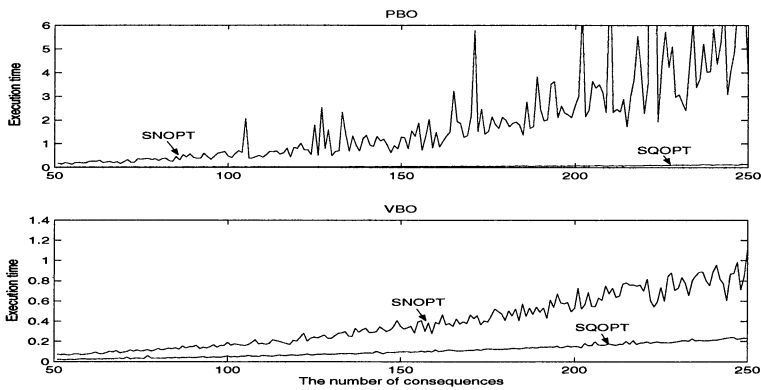


Fig. 3. Revised Active Set Method vs. PBO and VBO

Table 1. Numerical Results

Consequences	51-100	101-150	151-200	201-250
PBO SNOPT	0.3320	0.9098	1.9729	4.7723
SQOPT	0.0186	0.0311	0.0545	0.0981
Gap	17.8599	29.2718	36.2024	48.6389
VBO SNOPT	0.1111	0.2449	0.4460	0.7457
SQOPT	0.0330	0.0697	0.1216	0.1948
Gap	3.3658	3.5138	3.6660	3.8273

3.4 Impacts from the Number of Constraints

Strategy 1 In this strategy, we apply the simulated decision situation containing two alternatives, each alternative with $N = 150$ consequences (600 variables), and with around $2N$ (300) constraints. As for handling the number of constraints, we add them one by one, i.e., from 1 to $2N$. Decision situations with $N = 200$ and $N = 250$ consequences are applied analogously. This experiment applies to 25 groups of data. Each group consists of three instances containing 150, 200, 250 consequences, respectively. Correspondingly, a total of $(300 + 400 + 500) \times 25 = 30000$ simulated decision situations are computed.

Figure 4 shows the results of this strategy. We observe that for these three decision situations, the execution times fluctuate, showing no regular trend. Hence, the number of constraints could hardly impact upon the performance of SNOPT according to this strategy.

Strategy 2 In this strategy, full use is made of N , $2N$, $3N$, and $4N$ constraint sets, respectively. Each constraint set is applied directly this time, while the number of consequences increases from 51 to 250. This experiment applies to 25 groups of data. Each group consists of three instances from 51 to 250

consequences, respectively. Correspondingly, a total of around $200 \times 4 \times 25 = 20000$ simulated decision situations are computed. Figure 5 demonstrates the experimental results of this strategy.

Apart from the rise in execution times with the increase in the number of consequences, no clear dominance between these four constraint sets could be observed. Accordingly, the number of constraints could hardly affect the execution time of a general decision situation.

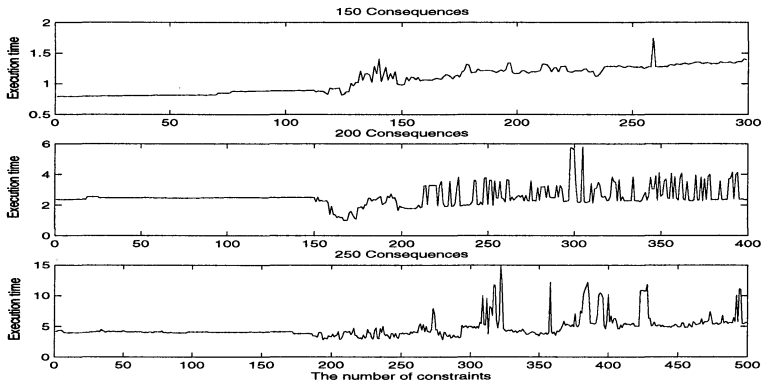


Fig. 4. Strategy 1

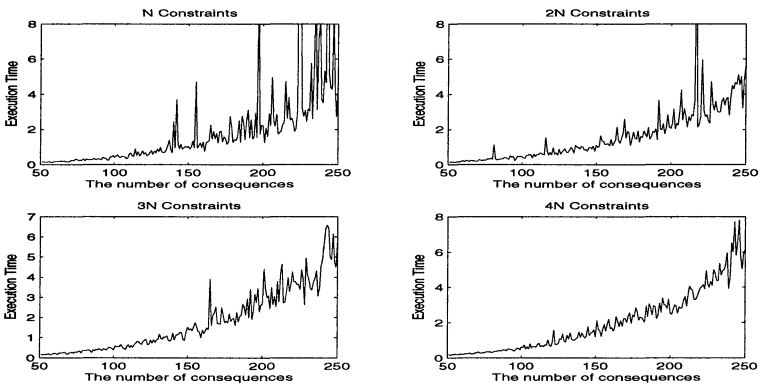


Fig. 5. Strategy 2

3.5 Impacts from Different Combinations

In this experiment, three strategies representing different combinations of the three types of constraints, as shown in Table 2, are taken. For each strategy, there are around $2N$ constraints. It applies to 25 groups of data. Each group consists of instances from 51 to 250 consequences, respectively. Correspondingly, a total of around $200 \times 3 \times 25 = 15000$ simulated decision situations are computed.

Table 2. Combination Strategies

	Comparative Difference Compound		
Strategy I:	50%	25%	25%
Strategy II:	50%	50%	0
Strategy III:	100%	0	0

Figure 6 demonstrates the results of this simulation study.

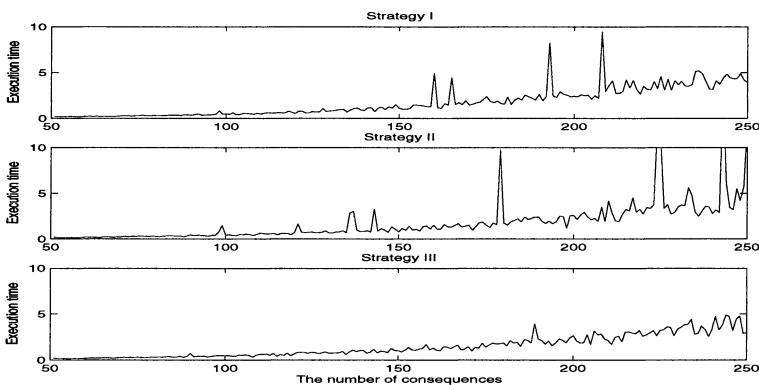


Fig. 6. Various Combinations

Although, intuitively a difference is supposed to exist, no obvious dominance could be observed. One reasonable explanation may lie in the 2 normalization constraints for *P-base*. Most software packages are able to successfully cope with the box constraint for a variable. Nevertheless, none are particularly designed to handle the normalization constraints. Not surprisingly, 2 normalization equations always remain internally as general constraints. Even if there were a StrategyIII containing 100% comparative statements, 2 normalization constraints would still make this strategy no different to others from a computational viewpoint.

According to this experiment, for a general decision situation evaluated by the DELTA method, different combinations of three types of constraint statements can hardly affect the execution time of SNOPT.

4 Conclusions and Future Work

In consistence with the results from the simulation study, the number of constraints and different types of constraints barely affect the execution time of SNOPT. A large gap exists between BLP and LP algorithms, which makes the idea of searching for a fast BLP algorithm possible. The difference between PBO and VBO could be a useful research direction to follow. General BLP algorithm can not meet the demand of an interactive software package.

In order to realize the new optimization algorithm, our future research work plans to begin with the revised active set method, and to take full advantage of the particular structure in (1) combined with certain advanced techniques from Sequential Quadratic Programming (SQP), which is adopted by the well-known NLP solver SNOPT.

References

1. Beale, E.M.L. (1955) On Minimizing a Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society, (Series B)* 17, 173–184.
2. Danielson, M. (1997) Computational Decision Analysis. Doctoral Thesis. Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology, Stockholm, Sweden.
3. Danielson, M., and Ekenberg, L. (1998) A Framework for Analysing Decisions under Risk. *European Journal of Operational Research*, Vol. 104/3, 474–484.
4. Danielson, M. (2004) Generalized Evaluation in Decision Analysis. To appear in *European Journal of Operational Research*.
5. Dantzig, G.B. (1963) Quadratic Programming. *Linear Programming and Extensions*, Princeton University Press, Princeton, USA, Chap. 12-4, 490–498.
6. Ekenberg, L. (2000) The Logic of Conflicts between Decision Making Agents. *Journal of Logic and Computation*, Vol. 10, No. 4, 583–602.
7. Ekenberg, L., Boman, M., and Linneroth-Bayer, J. (2001) General Risk Constraints. *Journal of Risk Research*, 4(1), 31–47.
8. Ekenberg, L., Brouwers, L., Danielson, M., Hansson, K., Johansson, J., Riabacke, A., and Vári, A. (2003) Simulation and Analysis of Three Flood Management Strategies. IIASA Internal Report, IR-03-003.
9. Fletcher, R. (2000) Stable Reduced Hessian Updates for Indefinite Quadratic Programming. *Mathematical Programming*, 87(2), 251–264.
10. Lemke, C.E. (1968) On Complementary Pivot Theory. In Dantzig, G.B., and Veinott, A.F. *Mathematics of Decision Sciences*, AMS, Providence, Rhode Island, Part 1, 95–114.
11. Pardalos, P.M., and Romeijn, H.E. (2002) *Handbook of Global Optimization* Vol. 2, Kluwer Academic Publishers, the Netherlands, Chap. 5, 151.
12. Wolfe, P. (1959) The Simplex Method for Quadratic Programming. *Econometrical* 27, 382–398.

knowCube – a Spreadsheet Method for Interactive Multicriteria Decision Making

Hans L. Trinkaus

Fraunhofer Institute for Industrial Mathematics, Department of Optimization,
Gottlieb-Daimler-Str. 49, D-67663 Kaiserslautern, Germany
trinkaus@itwm.fhg.de

Abstract. *knowCube* is a novel multicriteria decision support system, consisting of components for knowledge organization, generation, and navigation. Knowledge organization rests upon a data base for managing qualitative and quantitative criteria, together with add-on information. Knowledge generation serves filling the data base via e.g. identification, optimization, classification or simulation. For “finding needles in haystacks”, the knowledge navigation component supports graphical data base retrieval and interactive, goal-oriented problem solving. Navigation “helpers” are, for instance, cascading criteria aggregations, modifiable metrics, ergonomic interfaces, and customizable visualizations. Examples from real-life projects, e.g. in industrial engineering and in the life sciences, illustrate the application of *knowCube*.

1 Introduction

Usually, real-life decision problems are characterized by several criteria or objectives to be taken into consideration. Despite the fact that “multiple objectives are all around us”, as Zeleny [1] points out, the dissemination of methods for multicriteria decision making (MCDM) into practice can still be regarded as insufficient. A principal reason for that situation might be that practitioners are just not familiar with MCDM methods, which often are difficult to understand and to work at. One main difficulty for an easy and explorative usage of methods might be caused by the user interface, i.e. the handling of interactions with a corresponding computer program. Zionts [2] formulates this obvious necessity of user-friendly software as follows: “I strongly believe that what is needed is a spreadsheet type of method that will allow ordinary people to use MCDM methods for ordinary decisions.”

knowCube, an innovative multicriteria decision support system (MCDSS), mainly originates from focussing on such user-specific needs. Therefore, the first aspect of the system is the provision of visualizations and interactivity in categories well understandable by non-expert users. All technical stuff, necessary for this comfortable presentation of and access on knowledge, is hidden in the background, combined in a so-called navigation component. Its integration with further components, for generating and managing knowledge data required for decision processes, is another main feature of this MCDSS.

The paper gives a scope of *knowCube*, the emphasis lies on presenting real projects already done. More details are found in Hanne and Trinkaus [3].

2 Knowledge Generation, Organization, Navigation

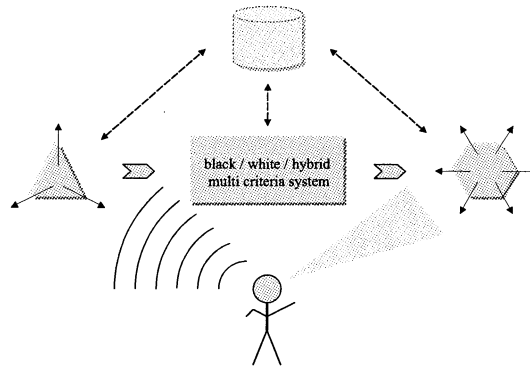


Fig. 1. “One picture means more than thousand words”

The sketch perfectly illustrates the scope of *knowCube*: A decision maker, comfortably interacting (by ear, eye, mouth, or hand) with an MCDSS!

Looking on the core of a multicriteria decision task, the following cases are to distinguish: If the description of the problem is known completely, it is called a “white” (box) multicriteria problem – as mostly assumed in optimization methods, applied for finding “the best of all possible solutions”. If nothing is known about the problem’s structure, it is denoted as “black”. Typically, outcomes of such systems are found by trial and error, or won by general inquiries – so they are frequently not reproducible or changing with time. Most real decision situations however are in between “black or white”, leading to “hybrid” problems. Then carrying out costly experiments and analyzing the results in combination with simulation methods are the means chosen for getting decision suggestions.

This central part of decision making: generation of knowledge, is highly specific, in most cases it must be customized according to the needs of acute problem settings. But, the other aspects may and should be supported by standardized tools. This was one starting point for the idea of *knowCube*.

Seen as a general framework of an MCDSS, it consists of three main components: *knowOrg*, *knowGen*, and *knowNav* (knowledge organization, generation, and navigation). These are put together with their sub modules into a common box, showing to various decision makers varied views, just like the different faces of a cube. This gives a hint at the term “knowledge cube”, abbreviated *knowCube*, taken as a “logo” for the complete MCDSS. Fig. 2 illustrates its structure, Hanne and Trinkaus [3] give additional technical details (especially on *knowOrg* and *knowNav*) – and so there’s some place left for explaining the more interesting special features of *knowCube* by two nice examples in the following sections.

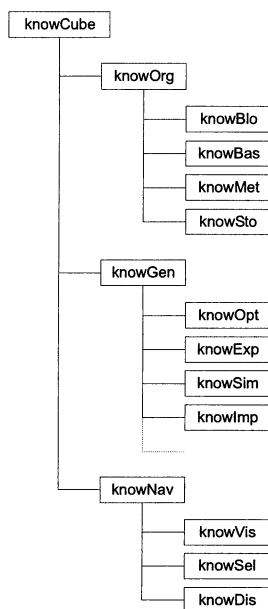


Fig. 2. The Structure of *knowCube*

3 Finding an Ideal Radiation Therapy Plan

Within the complete, time consuming workflow process of intensity modulated radiation therapy (IMRT), two challenging tasks emerge: First, the segmentation of CT (computer tomography) or MR (magnetic resonance) slides, i.e. adding curves onto the slides, describing the boundaries of the tumor and the neighboring organs at risk. Second, the dose calculation, i.e. computing the dose contributions to each voxel (volume element inside the body) of interest, coming from corresponding bixels at the beamheads (beam-head pixels) of a linear accelerator, used for the radiation treatment.

Some output of these two steps is taken as input for the knowledge data base: General information concerning the patient to be treated, his/her CT or MR slides, the segmentation contours, and the dose matrix.

Now the optimization procedure, done within the module *knowOpt*, starts, taking about four hours on a PC. It generates up to 1000 distinct Pareto optimal 3D-solutions, “covering” the planning horizon with a grid of alternatives for the patient. Here, a solution assigns a real number to each bixel, representing the amount of radiation emitted from there. The superposition of all bixel contributions sums up to a radiation distribution in the volume of interest, containing the tumor and the organs at risk. Each of these objects obtains a set of “notes” – as e.g. numbers, functions, and point sets – by which the decision maker will be able to estimate their “qualities”. (More details on IMRT are presented in Bortfeld [4] and Bortfeld et al. [5].)

All calculations mentioned above happen automatically, they may be done overnight as a batch job, and stored in the data base *knowBas*. The next day, the results may be transferred to a notebook, such that the physician (and even his/her patient) may look for the optimal plan interactively.

And then, managing this enormous quantity of data to find out the “best” plan with reasonable effort works by

- real-time accesses on the pre-calculated and -processed data base – which in particular contains various data aggregations in a cascading manner,
- simultaneous visualizations of dose distributions – by well known (for a physician, at least) dose-volume-histograms and colored isodose lines, superimposed on the patient’s grey-value CTs or MRs,
- the tricky exploration tool *knowCube Navigator*, supporting the interactive, goal oriented tour through the huge variety of solutions – leading towards the patient’s optimal radiation therapy plan within minutes.

Fig. 3 shows a part of the graphical user interface (GUI) at the beginning of a typical planning session, when the main pointer of the database references on the solution of least common tolerance – a kind of “average solution”, that balances “notes” of conflicting criteria.

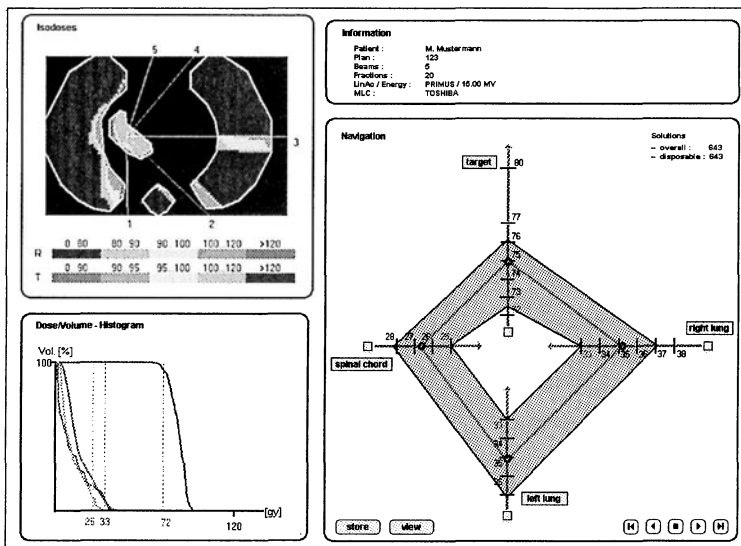


Fig. 3. GUI for IMRT

Information objects are presented there in a decreasing degree of fineness:

- “Richest” information is given by stacks of CTs or MRs through the neighborhood of the tumor (frontal, sagittal, and transversal – as selected here for the sketch), superimposed by in advance calculated isolines.

- “Medium” information content is aggregated in the dose-volume histogram, where each curve shows the accumulated Gray contribution per organ at risk or tumor. (Gray is the physical unit for radiation energy.)
- The “coarsest” information is delivered by the navigation object of the GUI: All radiation energy absorbed by an organ at risk – or by the tumor – is summarized in a corresponding number. These numbers are inscribed as pull points (i.e. grip points) on the organs’ or tumor’s acceptance intervals (i.e. decision ranges), arranged in a star-shaped sketch. Linking these grip points by line segments results in a solution polygon (i.e. navigation polygon), characterizing the whole planning solution from a top point of view, well suited for a first, quick estimate of the solution’s relevance.

All solutions together define the ranges of the acceptance intervals. Connecting their endpoints delimits the shaded area, called planning horizon (i.e. decision range). Navigating happens by a mouse-click on a pull point, moving it towards smaller/bigger values. The data base, indexed by modified lexicographic ordering strategies, is scanned simultaneously, isolines and dose-volume curves of “neighboring-solutions” are updated on the screen at once. As the mouse-click is released, the “moving polygon” stops in a new position. This first kind of action: pull, may be repeated to explore the planning horizon. But, due to Pareto optimality, going to a “better” Gray value on some axis must be paid by a “worse” value somewhere else. Here helps a second kind of action: lock. Again, only by a mouse-click in a lock-box, the planning horizon immediately is divided into differently shaded parts. Active navigation is restricted, a filter selects the accessible objects in the data base.

Some more actions facilitate the physician’s work: storing, viewing and skimming, by using well known recorder button functionalities. All of them support: Navigating in a planning horizon – towards an optimal solution!

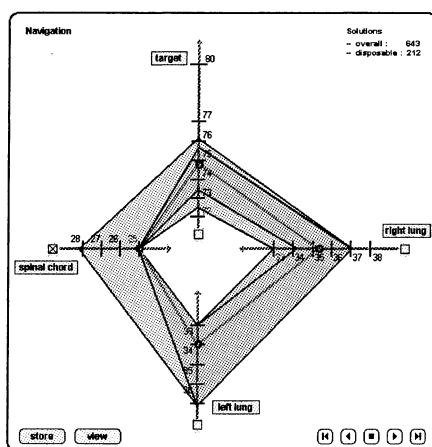


Fig. 4. A Navigation Tour – One Intermediate Step

4 Designing Best Trunking Devices

An engineering team got the task of designing the profile of a new electrical trunking device – a product usually fixed at the wall, and used for comfortable laying of multitudes of cables. Many objectives are to be taken into account in such a development process. Fig. 5 helps for a better understanding.

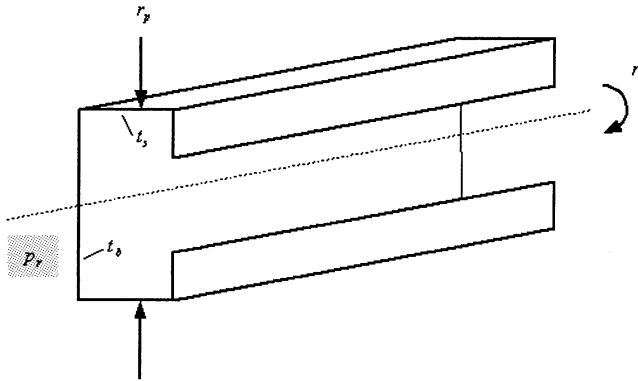


Fig. 5. Cross Section of a Trunking Device

r_p points to the resistance of the trunking device against pressing from opposite sides, r_t to its resistance against torsion, and p_r denotes the portion of recycling material (instead of pure PVC). These variables should have big values, of course. Together with t_b and t_s (the thickness of bottom and side faces) they characterize the “look and feel” of a specific design variant. Another group of variables represents production time, costs, and amount of material needed. All these eight variables are “working together”, in non-trivial, highly nonlinear, and partially conflicting combinations. (Some more technical background may be found in Ebeling [6].)

After designing plenty of different profile versions, evaluating them virtually (e.g. by some finite element methods out of *knowSim*), and putting all together into the data base, the job of the engineering team is done. Then the management has to decide which new profile should be launched – but, without looking over hundreds of data sheets.

Here the component *knowNav* helps again: A closed polygon within a star-shaped scale arrangement visualizes a data base record, i.e. a design variant. Other objects (e.g. acceptance intervals, decision horizon) and functionalities (e.g. locking, storing) are analogous to the IMRT case. Navigation happens via moving the polygon by mouse-clicks, in “extreme” cases also by the CEO. So, *knowCube* serves as an ideal base for discussions at meetings, where many decision makers are fighting with plenty, usually conflicting, arguments.

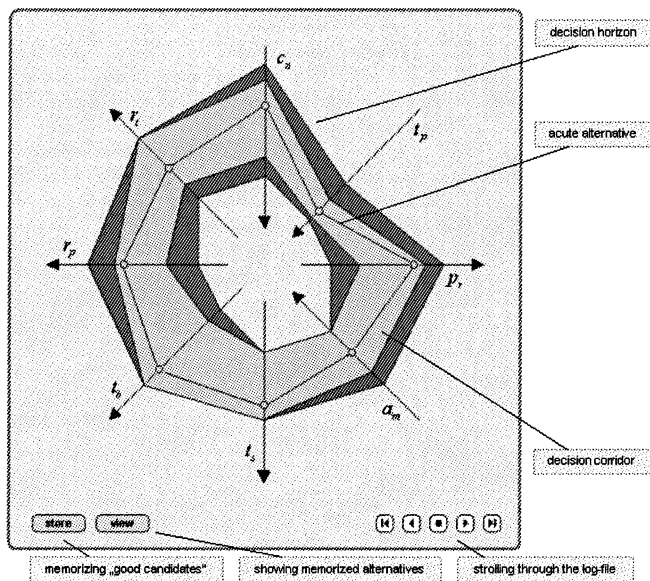


Fig. 6. *knowCube Navigator*

For this application, Fig. 6 shows some parts of the GUI, Fig. 7 presents popup-information, as e.g. the (exaggerated) torsion under a certain force, and the material stress visualized by colorings of distinguished profile details.

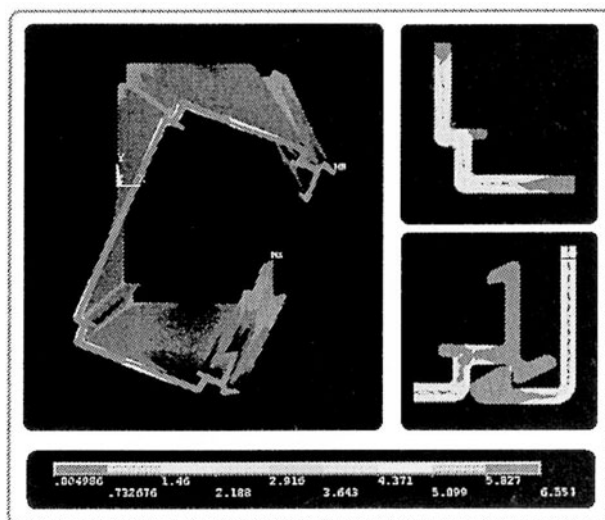


Fig. 7. Design of a Trunking Device – Information Attachment

5 Summary, Conclusions, and Outlook

knowCube was introduced, a novel interactive decision support system, integrating various tools for knowledge organization, generation, and navigation.

The main guideline for designing *knowCube* was to have a user-friendly visual interface, and to utilize interactivity in terms which are familiar to a non-expert decision maker, in particular for an "intuitive surfing through data bases of alternatives" – according to the already cited statement of Stanley Zionts: "I strongly believe that what is needed is a spreadsheet type of method that will allow ordinary people to use MCDM methods for ordinary decisions." I hope, that *knowCube* points towards this direction, and experiences won by the two application examples already realized in practise strongly are confirming this hope.

This paper was dealing only with quantitative criteria – to keep it simple, to introduce the main concepts of *knowCube*, as a first step, and to have place for explaining the basics by examples.

To work also with other types of criteria – like qualitative, objective or subjective, rational or irrational, active or passive, dependent or independent, deterministic or statistic, hard or soft, timely, ..., and all mixed together – the topics introduced so far must be extended. This will happen in a paper to follow. And again, the main new ideas therein will be motivated, guided and demonstrated by real-life applications.

6 Acknowledgements

Thanks a lot to my colleagues Heiko Andrä, Thomas Hanne, and Karl-Heinz Küfer for many valuable discussions, thanks to the Tehalit GmbH, Heltersberg, Germany, for setting and funding the trunk device optimization task, and thanks to Thomas Bortfeld at Massachusetts General Hospital, Boston, USA, for the cooperation in intensity modulated radiation therapy.

References

1. Zeleny, M. (1982): Multiple Criteria Decision Making, McGraw-Hill, New York.
2. Zionts, S. (1999): Foreword. in Gal, T., Stewart, T.J., Hanne, T. (eds.): Multicriteria Decision Making - Advances in MCDM Models, Algorithms, Theory, and Applications. Kluwer Academic Publishers, Boston/Dordrecht/London.
3. Hanne, T., Trinkaas, H.L. (2003): *knowCube* for MCDM – Visual and Interactive Support for Multicriteria Decision Making. Berichte des ITWM, Nr. 50.
4. Bortfeld, T. (1995): Dosiskonformation in der Tumorthherapie mit externer ionisierender Strahlung. Habilitationsschrift, Universität Heidelberg.
5. Bortfeld, T., Küfer, K.-H., Monz, M., Scherrer, A., Thieke, C., Trinkaas, H.L. (2003): Intensity-Modulated Radiotherapy: A Large Scale Multi-Criteria Problem. Berichte des ITWM, Nr. 43.
6. Ebeling, F.-W. (1974): Extrudieren von Kunststoffen. Vogel-Verlag, Würzburg.
7. Wickelgren, W.A. (1979): Cognitive Psychology, Prentice-Hall, Englewood Cliffs.